

**LAPORAN KERJA PRAKTEK
PT. ATNIC EKOTEKNO WICAKSANA**



**Dipersiapkan Oleh :
Dewa Gede Agung Ari Septiyadi / 150708241**

**Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta
2019**

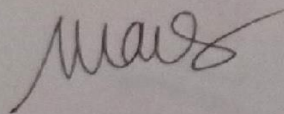
HALAMAN PENGESAHAN

Laporan Kerja Praktek

Laporan ini telah diperiksa dan disetujui
Pada tanggal 25 September 2019

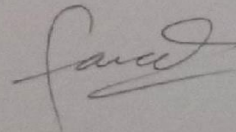
Oleh:

Dosen Pembimbing,



(Martinus Maslim, S.T., M.T.)

Pembimbing Lapangan,



(Farid Inawan, S.T.)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Ida Sang Hyang Widhi Wasa, Tuhan Yang Maha Esa, yang senantiasa telah memberikan rahmat dan anugerah-nya, sehingga penulis dapat menyelesaikan Laporan kerja Praktek di PT. Atnic Ekotekno Wicaksana ini. Kerja Praktek merupakan sebuah mata kuliah yang ada di dalam kurikulum Strata 1 Teknik Informatika universitas Atma Jaya Yogyakarta. Kerja Praktek mempunyai beban studi sebesar 2 SKS. Kerja Praktek ini merupakan syarat bagi mahasiswa untuk melanjutkan mengerjakan Tugas Akhir. Kerja Praktek ini memberikan pengalaman kerja di perusahaan secara langsung kepada mahasiswa.

Melalui Kerja Praktek ini, mahasiswa diharapkan untuk bisa mendapatkan pengalaman kerja secara langsung di perusahaan dan mengetahui tentang lingkungan kerja, proses bisnis, dan teknologi informasi yang digunakan pada perusahaan. Mahasiswa juga diharapkan untuk dapat menerapkan ilmu yang didapatkan selama perkuliahan untuk dapat mengerjakan project dan tugas yang diberikan selama Kerja Praktek. Laporan ini dibuat berdasarkan pengalaman yang didapatkan dari Kerja Praktek yang telah ditempuh selama 2 bulan sejak tanggal 2 Juli 2019 sampai 2 Agustus 2019 di PT. Atnic Ekotekno Wicaksana

Pelaksanaan Kerja Praktek ini dapat diselesaikan dengan baik tidak terlepas dari bantuan beberapa pihak, oleh karena ini penulis hendak mengucapkan terimakasih kepada:

1. Ida Sang Hyang Widhi Wasa, Tuhan Yang Maha Esa yang selalu menyertai dan memberikan penyertaan dan anugrahnya kepada penulis.
2. Orangtua yang selalu mendukung dan memberikan doa dan semangat kepada penulis selama pelaksanaan Kerja Praktek.
3. Bapak Martinus Maslim, S.T., M.T, selaku dosen pembimbing yang telah membimbing penulis selama Kerja Praktek dan penyusunan laporan.
4. Mas Farid Inawan selaku *Software Lead*, dan juga pembimbing lapangan, mas Dama selaku Senior Android Developer, Mas Adhya selaku Senior Web Developer dan Mas Syauqi Nurul Aziz selaku *CTO*, yang telah membimbing, membantu dan memberikan banyak masukan dan pengalaman kepada penulis selama pelaksanaan Kerja Praktek.
5. Seluruh anggota tim PT. Atnic Ekotekno Wicaksana yang telah memberikan banyak pengalaman dan masukan kepada penulis selama Kerja Praktek ini.

6. Kadek Devi yang selalu menasehati dan memberikan semangat dan dukungan kepada penulis selama pelaksanaan Kerja Praktek.
7. Dan kepada seluruh teman-teman yang telah membantu dan memberikan masukan kepada penulis selama pelaksanaan Kerja Praktek.

Akhir kata penulis menyadari bahwa pelaksanaan Kerja Praktek dan Penyusunan Laporan ini masih jauh dari kata sempurna. Oleh karena itu kritik dan saran sangat penulis harapkan untuk perkembangan penulis di masa depan, semoga laporan ini dan bermanfaat bagi semua pihak-pihak yang membutuhkan. Terimakasih

Yogyakarta

2019

Dewa Gede Agung Ari Septiyadi

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN.....	1
1.1 Sekilas PT. Atnic Ekotekno Wicaksana.....	1
1.2 Sejarah singkat PT. Atnic Ekotekno Wicaksana.....	1
1.3 Visi dan Misi PT. Atnic Ekotekno Wicaksana.....	2
1.4. Struktur Organisasi PT Atnic Ekotekno Wicaksana.	4
1.5. Deskripsi Tugas dari Struktur Organisasi PT.Atnic Ekotekno Wicaksana.....	5
1.6 Departemen IT dalam Perusahaan PT. Atnic Ekotekno Wicaksana	6
BAB II PELAKSANAAN KERJA PRAKTEK	7
2.1. Penjelasan Logbook	7
2.2 Hasil Pekerjaan Secara Umum.....	41
2.3. Bukti Hasil Pekerjaan.....	42
BAB III HASIL KERJA PRAKTEK.....	53
3.1 Manfaat Kerja Praktek	53
3.2 Penerapan Ilmu Dalam Kerja Praktek.....	54
BAB IV KESIMPULAN.....	56

DAFTAR GAMBAR

Gambar 1.1 Struktur Organisasi Perusahaan PT. Atnic Ekotekno Wicaksana	4
Gambar 2.1 Tampilan <i>Homepage</i> Fitur <i>Credits</i>	42
Gambar 2.2 Fitur Tambah Data <i>Credits</i>	43
Gambar 2.3 Fitur <i>Edit Data Credits</i>	43
Gambar 2.4 Tampilan <i>Homepage</i> Fitur <i>Courses</i>	44
Gambar 2.5 Fitur Tambah Data <i>Courses</i>	45
Gambar 2.6 Fitur <i>Edit Data Courses</i>	45
Gambar 2.7 Tampilan <i>Homepage</i> Fitur <i>Periode</i>	46
Gambar 2.8 Fitur Tambah Data <i>Periode</i>	46
Gambar 2.9 Fitur <i>Edit Data Periode</i>	47
Gambar 2.10 Tampilan <i>Homepage</i> Fitur <i>Study</i>	48
Gambar 2.11 Tampilan <i>Homepage</i> Fitur <i>Peran</i>	49
Gambar 2.12 Tampilan <i>Homepage</i> Fitur <i>Hak Akses</i>	49
Gambar 2.13 Tampilan <i>Homepage</i> fitur <i>Parameter</i>	50
Gambar 2.14 Tampilan <i>Homepage</i> Fitur <i>Alat</i>	51
Gambar 2.15 Tampilan <i>Homepage</i> Fitur <i>Kategori Produk</i>	51

BAB I

PENDAHULUAN

1.1 Sekilas PT. Atnic Ekotekno Wicaksana

PT. Atnic Ekotekno Wicaksana merupakan sebuah perusahaan yang bergerak dalam bidang agrikultur dan akuakultur, khususnya dalam bidang peternakan udang, udang merupakan salah satu produk yang menguntungkan, namun banyak masalah yang dialami dalam budidaya udang. Mulai dari masalah kualitas air yang mengarah ke munculnya penyakit, kondisi cuaca yang buruk, tenaga kerja atau petambak yang kurang terampil, dan berbagai masalah yang dialami dalam manajemen pertambakan udang.

Berangkat dari masalah tersebut PT. Atnic Ekotekno Wicaksana membuah sebuah perangkat yang bernama JALA. JALA merupakan perangkat *Internet of Things (IoT)* yang memungkinkan petambak udang untuk mengetahui kondisi air tambaknya. Perangkat JALA dilengkapi dengan berbagai sensor yang dapat membantu untuk mengetahui kondisi air, seperti sensor oksigen terlarut, pH, salinitas dan *TDS (Total Disolved Solid)*. Kondisi air tambak tersebut kemudian akan dilaporkan melalui sebuah aplikasi berbasis WEB dan *Mobile* yang dilengkapi dengan sistem pendukung keputusan untuk membantu petambak udang mengatur tambaknya dan meningkatkan respon petambak dalam menjaga kualitas air serta mengurangi kesalahan penanganan pada udang.

1.2 Sejarah singkat PT. Atnic Ekotekno Wicaksana

PT. Atnic Ekotekno Wicaksana, yang pada mulanya bernama Atnic, didirikan pada tahun 2014. Atnic dimulai dengan menerima berbagai pesanan untuk membuat berbagai macam perangkat teknologi secara spesifik seperti jam digital dengan fitur yang unik, lampu kubus dengan animasi LED, desain LED dan lain sebagainya.

Pada tahun 2014, Atnic mendapatkan tawaran dari PT Indmira untuk berkolaborasi membuat perangkat teknologi berdasarkan masalah-masalah yang ada. Kemudian, Tim Atnic melakukan reset dan menciptakan *prototype* pertama mereka pada tahun 2015. Atnic menciptakan JALA perangkat *Internet of Things(IoT)* yang memungkinkan petani tambak udang untuk melakukan kontrol pada tambak udangnya.

Pada maret 2015, Jala memenangkan kompetensi dari ASME (*The American Society of Mechanical Engineers*). Kemudian pada bulan April ditahun yang sama, Atnic mulai menjalankan program inkubasi dari INOTEK. Pendanaan dari dua belah pihak dimanfaatkan untuk penelitian dan pengembangan yang dimulai pada bulan Juli 2015 dengan 4 pendiri yang tersisa pada bulan November 2015, dengan bantuan PT. Indmira, Atnic secara resmi berganti menjadi PT. Atnic Ekotekno Wicaksana, sebagai perusahaan baru yang memberi solusi teknologi terhadap agrikultur dan akuakultur.

1.3 Visi dan Misi PT. Atnic Ekotekno Wicaksana

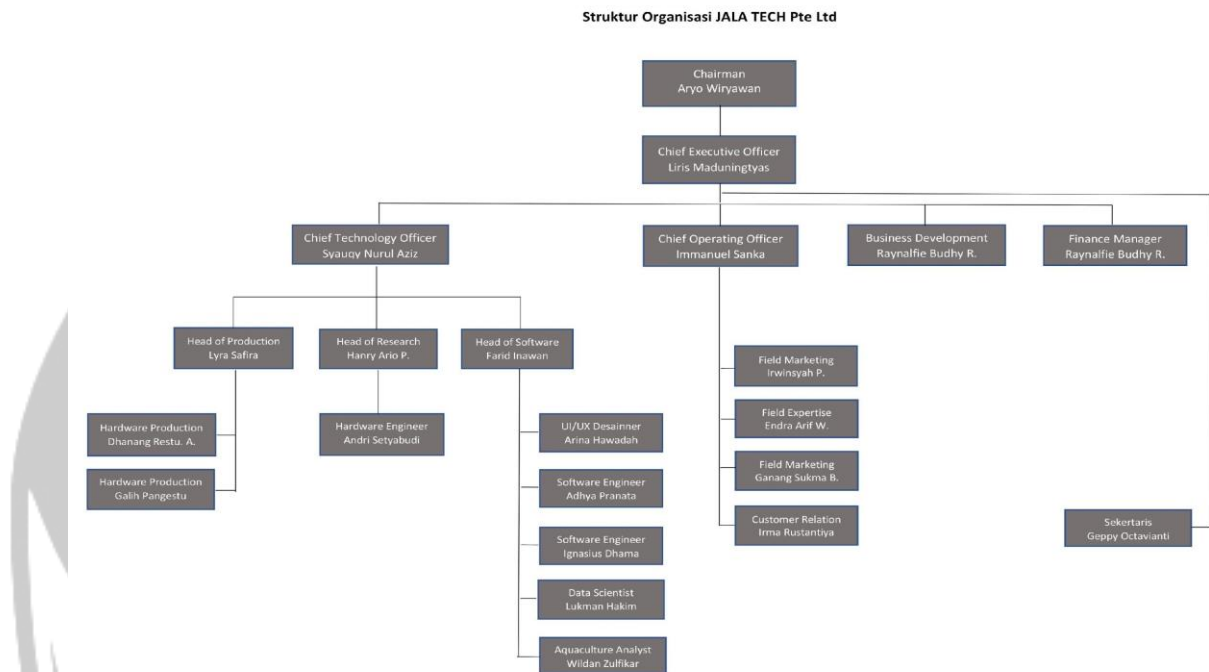
Visi

- *Technology Research and Development (R&D)* We continuously conduct R&D and train our expert staff to support consultation and technology monitoring services
- *Eco Friendly Products* Our products are made from renewable materials and safe for the environment, and manufactured in a factory that uses zero emission concept.
- *Responsible Ponds Use* Our production ponds and those of our partners use a sustainable aquaculture concept with well managed environment.

Misi

- *Company Based on Technology R&D* PT. Atnic Ekotekno Wicaksana is equipped with a research facilities that are always updated to improve formulation and apply products. PT. Atnic Ekotekno Wicaksana owns field research facilities to develop technologies and production management. With those technological guarantees, PT. Atnic Ekotekno Wicaksana products will always be up to date and in step with costumers' needs
- *Technology-Enabled Customized Products* Our manufacturing process is designed to accommodate customers' needs, and produce customized products up to a certain scale.
- *Consultation and Supervision* The scope of our consultation and supervision includes the technology and management of aquaculture. We can analyze the water and other parameters to determine the right technology and offer periodical supervision

1.4. Struktur Organisasi PT Atnic Ekotekno Wicaksana.



Gambar 1.1 Struktur Organisasi Perusahaan PT. Atnic Ekotekno Wicaksana

1.5. Deskripsi Tugas dari Struktur Organisasi PT.Atnic Ekotekno Wicaksana

1. *Chairman*

Memimpin rapat umum, dalam hal: untuk memastikan pelaksanaan tata-tertib, keadilan dan kesempatan bagi semua untuk berkontribusi secara cepat menyesuaikan alokasi waktu per item masalah, menentukan urutan agenda, mengarahkan diskusi kearah konsensus, menjelaskan dan menyimpulkan tindakan dan kebijakan. sebagai perwakilan organisasi dalam hubungan dengan dunia luar.

2. *CEO*

Merencanakan, mengelola, dan menganalisis segala aktivitas fungsional bisnis seperti operasional, sumber daya manusia, keuangan, dan pemasaran.

3. *CTO*

Menyatukan pengembangan produk, memahami perkembangan teknologi saat ini dan penguadopsiannya, dan mengelola pengembangan produk.

4. *COO*

Bertanggung jawab untuk merancang proses marketing dan mengontrol perkembangan client di lapangan.

5. *Head of Software*

Bertanggung jawab dalam memimpin dan mengontrol pengembangan software untuk JALA

6. *Head of Research*

Bertanggung jawab dalam memimpin dan mengontrol pengembangan software untuk JALA

7. *Head of Production*

Bertanggung jawab dalam memimpin dan mengontrol perkembangan produk dan pengelolaan produk

8. *Business Development*

Bertanggung jawab dalam mengetahui perkembangan pelanggan, perkembangan pasar yang akan ditargetkan, dan menjalin relasi hubungan dengan perusahaan luar.

9. *Finance Manager*

Bertanggung jawab untuk membantu perencanaan bisnis serta mengambil keputusan dalam mengelola keuangan yang sesuai dari perusahaan.

10. Sekretaris

Bertanggung jawab untuk memberikan informasi kepada pimpinan untuk menjalankan rapat yang akan dilaksanakan. Bertanggung jawab untuk mengurus surat-surat yang akan dikeluarkan ataupun yang akan diberikan oleh perusahaan.

1.6 Departemen IT dalam Perusahaan PT. Atnic Ekotekno Wicaksana

Department IT dalam PT Atnic Ekotekno Wicaksana, terbagi menjadi 3 yaitu *Head Of Production, Head Of Research, Head Of Software*. *Department Head Of Production* lebih memusatkan tugasnya pada produk yang akan diproduksi maupun yang akan dibeli. Sedangkan *Department Head Of Software* lebih memusatkan tugasnya pada pengembangan aplikasi JALA berbasis Web dan Mobile serta pengembangan perangkat lunak yang diaplikasikan pada perangkat Jala. Sedangkan *Department Head Of Research* memusatkan tugasnya pada pengembangan perangkat Jala agar dapat digunakan dan diimplementasikan sebagai alat kontrol kualitas air pada tambak udang.

BAB II

PELAKSANAAN KERJA PRAKTEK

2.1. Penjelasan Logbook

Selasa 2 Juli 2019 (08.30 – 16.40)

Pada hari pertama saya berbincang-bincang dengan anak-anak *internship* yang ada di JALA, lalu pukul 10.30 saya diajak rapat mingguan oleh tim *software* di JALA, yang terdiri dari mas Farid Inawan sebagai Software Leader di JALA, mas Syauqi Nurul Aziz sebagai *CTO-Co-Founder* JALA, mas Adhya Pranata Sakti sebagai *Software Developer* Senior di JALA, mas Dama sebagai *Android Developer* dan *Front-End Developer*, mas Jihad sebagai *Software Developer* Junior dan mbak Arin sebagai *UI/UX* JALA, pada rapat saat itu saya berkenalan dengan seluruh anggota tim *software* JALA. Pada rapat ini juga saya berbincang-bincang dengan mas Farid Inawan sebagai *Software Leader* JALA, dan mas Syauqi Nurul Aziz sebagai *CTO-Co-Founder* JALA. Saya berbincang-bincang seputar skill coding yang saya miliki, dan tujuan saya Kerja Praktek di JALA, lalu pada pukul 13.00, saya diberikan beberapa modul tentang belajar *Laravel* oleh mas Farid. yang dimana *Laravel* merupakan *framework* dengan menggunakan bahasa pemrograman **PHP** yang dikembangkan dan dirilis di bawah lisensi **MIT** dengan konsep *MVC* (*Model, View Controller*). Lalu saya coba mengerjakan modul 1 yaitu tentang *Model, View, Controller* pada *Laravel*. Yang dimana modulnya diberikan dalam bentuk power point sehingga mudah dipahami. Langkah pertama yang saya lakukan yaitu menginstall **composer** yang dimana *composer* merupakan *package manager* untuk bahasa pemrograman **PHP** yang menyediakan format standar untuk mengelola dependensi **PHP**, untuk menginstall *composer* di **ubuntu** saya menggunakan syntax `sudo apt-get install composer`, lalu saya menginstall *Laravel 5.4*, karena di modul yang saya pelajari menggunakan *Laravel 5.4*, untuk menginstall *Laravel 5.4* saya menggunakan syntax `composer create-project --preper-dist`

`laravel/laravel=>5.4.* learn-mvc` yang dimana syntax tersebut digunakan untuk menginstall *framework Laravel 5.4* dengan nama ***learn-mvc***, lalu saya melanjutkan langkah-langkahnya sesuai dengan materi yang diberikan pada modul.

Rabu 3 Juli 2019 (09.00 – 17.05)

Pada hari kedua saya melanjutkan tentang latihan *Laravel* yang diberikan pada hari sebelumnya. Setelah selesai mas farid memeriksa pekerjaan saya sekaligus menanyakan tentang hal yang kurang saya mengerti, lalu saya menanyakan tentang *Validate* dan token di *Laravel*, lalu mas farid menjelaskan tentang *Validate* dan token di *Laravel* merupakan sebuah validasi *form* input untuk menghindari kerusakan aplikasi dengan metode-metode ***hacking***, karena sebelumnya saya sudah pernah mempelajari sedikit tentang dasar-dasar *Laravel*, jadi disini saya mengikuti seperti modul yang telah diberikan. Setelah selesai saya diberikan latihan lagi masih seputaran tentang *Laravel*, yaitu tentang *Http-Layer* pada *Laravel*. Yang mempelajari tentang *view*, *routing* dan juga *validation*, yang dimana langkah pertama yang saya lakukan yaitu membuat *project* baru dengan nama ***Http-Layer***, dengan menggunakan syntax `composer create-project --preper-dist laravel/laravel=>5.4.* http-layer`, lalu saya membuat satu routing di *Laravel* yang dimana routing adalah pendefinisian mapping ***URL*** yang diakses dengan aksi yang dilakukan, contoh routing yang tersedia di *Laravel* yaitu `Route::get($usri, $callback)`, di *Laravel* ada beberapa routing yang tersedia selain ***Get***, juga ada ***post, patch, delete, push*** dan ***option***. Untuk modul yang diberikan ini sangat bermanfaat bagi saya, karena saya dapat banyak mempelajari hal-hal baru seputaran *Laravel* yang belum saya ketahui sebelumnya. Seperti perbedaan routing dan *sessions*. Untuk *sessions* disini dimaksudkan untuk tempat penyimpanan sementara yang dimana juga digunakan untuk membedakan *client* yang sedang mengakses server, untuk di *Laravel*, *sessions* bertahan selama 2 jam dari pertama kali anda mengaksesnya.

Kamis 4 Juli 2019 (09.00 – 17.00)

Pada hari ketiga hampir sama dengan hari-hari sebelumnya yaitu saya melanjutkan latihan seputar *Laravel* yang telah diberikan, setelah selesai mas Farid melakukan pemeriksaan terhadap apa yang sudah saya kerjakan. Saya juga menanyakan beberapa hal, terutama tentang routing dan juga tampilan *front-end* yang sebelumnya sudah dibuatkan oleh mas Farid, dan saya tinggal menggunakan. Disini saya menjelaskan sedikit penulisan untuk routing pada *Laravel*, yaitu dengan menuliskan *code* pada file *route.php* di *Laravel* dengan menggunakan syntax `Route::get('bar/{id}', 'TestController@bar');` yang dimana kegunaan dari *code* tersebut yaitu kita memanggil satu *url* yaitu *bar/id*, dengan menggunakan *method get* yang dimana disini kita memanggil fungsi yang bernama *bar*, di file *TestController.php* dan untuk tampilan *front-end* disini yaitu tampilan *view* atau antarmuka sistem dengan user, disini sudah diberikan tampilan *view*-nya yang sudah menyesuaikan nama-nama variabel yang saya gunakan di *back-end*-nya jadi saya tinggal *copy paste* tampilan *view* yang diberikan ke folder *view* yang berada di *Laravel*. Setelah itu saya memeriksa semua *code* yang telah saya buat terutama yang ada di *controller* yang dimana pada *controller* saya membuat beberapa fungsi *CRUD* yang sesuai dengan modul, setelah sesuai saya mencoba untuk menjalankan *code* yang telah saya buat dengan menjalankan perintah `php artisan serve` yang dimana syntax tersebut digunakan untuk menjalankan *code* yang telah saya buat di *Laravel* yang dimana syntax tersebut langsung memberikan arahan *port* untuk kita mengakses *Url* pada *browser*, yang dimana *port default*-nya adalah 8000.

Sabtu 6 Juli 2019 (08.50 – 15.00)

Pada hari keempat saya diberikan latihan tentang Autentifikasi dan Otorisasi pada *Laravel*, yang dimana saya sebelumnya belum pernah mengenal istilah tersebut pada *Laravel*, dan langkah pertama yang saya lakukan yaitu menginstall *Laravel* dengan menggunakan `syntax composer create-project --preper-dist laravel/laravel=>5.4.* learn-auth`. Setelah saya menginstall *Laravel* lalu saya menjalankan `syntax php artisan make:auth`, yang dimana `syntax` tersebut digunakan untuk membuat sebuah autentifikasi pada *Laravel*, setelah saya menjalankan `syntax` tersebut maka *Laravel* otomatis membuat beberapa file di `views/auth`. Setelah itu saya menjalankan `syntax php artisan migrate`, yang dimana `syntax` tersebut digunakan untuk men-*generate* tabel-tabel yang sudah disediakan di *Laravel* ke database yang sudah kita buat sebelumnya, lalu saya mencoba untuk menjalankan *code* yang telah saya buat dengan menggunakan `syntax php artisan serve`, setelah itu saya mencoba untuk register dan *logIn* pada *Laravel* untuk memastikan bahwa sistem autentifikasi yang saya buat sudah berjalan dengan lancar, lalu saya diminta untuk menjalankan fungsi *forgot password* pada *Laravel*, dan setelah saya mencoba saya menemukan *error*, yang dimana sistem yang kita buat belum ada settingan *email*, jadi disini saya memanfaatkan sistem ***mailtrap.io*** yang digunakan untuk melakukan testing *email* tanpa perlu untuk melakukan *spamming* ke *email* asli. Setelah saya ikuti modulnya dan pelajari tentang *code-code* nya saya mulai memahami tentang autentifikasi dan otorisasi, autentifikasi dan otorisasi di *Laravel* digunakan untuk membuat fungsi *logIn*, *logOut*, register dan *forgot password* yang sudah tersedia di *Laravel*, jadi disini kita hanya menjalankan `syntax php artisan make:auth` maka semua fungsi tersebut otomatis ter-*generate* langsung. Tetapi disini saya masih banyak menanyakan beberapa hal kepada mas Farid, yaitu tentang penggunaan ***mailtrap.io*** serta fungsi dan juga kegunaanya pada *Laravel*. Dan setelah selesai mas Farid melakukan pemeriksaan terhadap latihan yang telah saya kerjakan.

Senin 8 Juli 2019 (08.40 – 16.30)

Pada hari kelima saya diberikan latihan tentang pembuatan sistem stok sederhana yang dimana latihan ini sudah mulai kompleks dan juga sudah menggunakan database, saya memulai pengerjaan dengan melakukan instalasi *Laravel* dengan menggunakan syntax `composer create-project --preper-dist laravel/laravel=>5.4.* learn-stock`. Setelah selesai saya membuat model dan *migration* tabel dengan menggunakan syntax `php artisan make:model Product` dan mengulangi langkah yang sama sesuai dengan materi yang diberikan, lalu saya membuat beberapa *migration* tabel dengan menggunakan syntax `php artisan make:migration --create=products create_products_tabel`, yang dimana syntax tersebut digunakan untuk membuat *migration* tabel dengan nama ***Products***, lalu mengulangi langkah yang sama sesuai dengan materi yang diberikan, setelah itu saya membuat *field-field* pada tabel yang sesuai dengan di materi yang diberikan, lalu saya menjalankan syntax `php artisan migrate` untuk men-*generate* tabel yang telah saya buat ke dalam database setelah itu saya membuat *controller* yang akan digunakan dengan menggunakan syntax `php artisan make:controller ProductController -r`, dan mengulangi langkah yang sama sesuai dengan materi yang diberikan, pada pembuatan *controller* saya menambahkan *flag* `'-r'` yang dimana *Laravel* akan otomatis men-*generate* beberapa *method resource* pada *controller* yaitu, *index*, *create*, *show*, *store*, *edit*, *update* dan *delete*. Setelah itu saya melanjutkan untuk mengikuti syntax yang ada pada materi, untuk tampilan *front-end* nya karena sudah diberikan jadi saya tinggal meng *copy* tampilan *view* nya ke folder *view* yang ada pada *Laravel*, setelah selesai saya mencoba untuk menjalankan *code*-nya. Setelah selesai saya menemukan beberapa hal yang kurang saya pahami, yaitu tentang ***Carbon*** dan Relasi database pada *Laravel*. Lalu mas Farid menjelaskan hal-hal yang saya tanyakan yaitu tentang *carbon*, *Carbon* di *Laravel* merupakan sebuah *library* untuk *date* (tanggal dan waktu) yang digunakan di *Laravel*. Setelah selesai

mas farid melakukan pemeriksaan dan saya menanyakan tentang struktur *code* yang digunakan serta pemanggilan model pada *controller*.

Selasa 9 Juli 2019 (09.00 – 17.00)

Pada hari keenam saya diberikan suatu *project* yaitu tentang membuat sebuah Sistem Kampus Sederhana, yang dimana pada *project* ini saya sudah diberikan desain database dengan analisa kerja sistem yang sudah diberikan, hari pertama saya mulai menginstall *Laravel* dengan menggunakan syntax `composer create-project --preper-dist laravel/laravel=>5.4.* kampus`. Setelah itu saya mulai membuat *migration* tabel pada *Laravel* dengan menggunakan syntax `php artisan make:migration --create=credits create_credits_tabel`, dan mengulangi langkah yang sama untuk semua tabel yang sudah diberikan pada soal, lalu saya melanjutkan untuk mengisi *field-field* yang ada pada tabel dan menjalankan syntax `php artisan migrate` untuk men-*generate* tabel yang telah saya buat ke database setelah itu saya membuat model dan *controller* untuk tabel ***credits*** dengan menggunakan syntax `php artisan make:model Credit` untuk model *Credit* dan `php artisan make:controller CreditController -r` untuk *Credit controller*, langkah pertama yang saya lakukan yaitu membuat satu fungsi *users* yang digunakan untuk merelasikan tabel *credit* dengan tabel *user* yang dimana disini memanfaatkan fungsi ***Eloquent*** yang tersedia pada *Laravel*, lalu saya menambahkan deklarasi variabel tabel dan *fillable* dengan menggunakan syntax `protected $table = 'credits';` yang dimana data akan dikirim ke tabel *Credits* lalu untuk *fillable* dengan menggunakan syntax `protected $fillable = ['user_id', 'credit'];` yang dimana data yang dikirim berupa *user_id* yang diambil dari tabel *user* dengan fungsi *eloquent* yang sudah kita buat, lalu data *credit* yang akan di tampung di *field* *credit* pada tabel *credit*. Lalu saya melanjutkan untuk membuat sebuah fungsi pada *controller index* pada *controller* setelah itu saya membuat routing pada file *route.php* di *Laravel* dengan menggunakan snytax `Route::get('/index',`

`'CreditController@index'');` yang dimana syntax tersebut digunakan untuk menampilkan data dari fungsi *index* di *controller*. Setelah itu saya mulai mengerjakan tampilan *website*-nya, dengan menggunakan *dashboard* dari *template bootstrap*, yaitu *template AdminLTE*. Setelah selesai saya melanjutkan untuk membuat satu *CRUD* pada tabel *Credits*, setelah selesai saya menanyakan tentang hasil pekerjaan saya dengan mas Farid, yaitu tentang alur sistem pada tabel *Credits*. Lalu saya melanjutkan untuk membuat *CRUD* pada tabel selanjutnya.

Rabu 10 Juli 2019 (09.20 - 17.30)

Pada hari ke tujuh saya melanjutkan mengerjakan *project* sebelumnya, yaitu tentang membuat autentifikasi pada sistem, yang dimana disini saya menggunakan **Role Multiple Login** pada *Laravel*, *role multiple login* merupakan sebuah sistem *logIn* untuk membatasi hak akses user ke beberapa fitur yang ada di *website*. Yang dimana setiap *user* yang bertipe berbeda yang dapat mengakses dan mengedit beberapa data sesuai dengan tipe *role* yang sudah diberikan pada tabel *users*. Untuk membuat *role multiple logIn* di *Laravel* saya terlebih dahulu menambahkan *field role* pada tabel *user*, untuk menampung data *role* yang akan diisi, lalu saya menjalankan syntax `php artisan make:auth` untuk membuat sistem *logIn* pada *Laravel*, seperti yang saya jelaskan sebelumnya. Lalu saya menambahkan fungsi *checkRole* pada folder *middleware* dengan menggunakan syntax `php artisan make:middleware checkRole`, yang dimana pada file *checkRole.php* akan otomatis dibuatkan fungsi *handle* untuk meng-*handle* *role* user yang *login*, disini saya sedikit memodifikasi fungsinya dengan menambahkan parameter `...$roles` pada fungsi *handle* yang dimana parameter tersebut digunakan untuk menampung *role* yang *logIn* dan untuk `...` di depan variabel `$roles` artinya untuk kita bisa menampung banyak jenis *role* yang *logIn*. Setelah itu saya membuat route group yang digunakan untuk membedakan *role* user yang bisa mengakses beberapa route yang ada sesuai dengan ketentuan soal yang diberikan, untuk membuat route group saya menggunakan *middleware* yang

sebelumnya sudah saya buat, dengan menuliskan *code*

```
Route::group(['middleware' => ['auth', 'checkRole:admin,mahasiswa']],function( ){} yang dimana code tersebut
```

digunakan untuk membuat route group dengan memanfaatkan *middleware* yang telah kita buat, dan memanggil fungsi *checkRole* yang dimana hanya admin dan mahasiswa yang bisa mengakses route yang ada didalamnya. Setelah selesai mas Farid melakukan pemeriksaan dan saya menanyakan beberapa hal tentang autentifikasi khususnya, yang saya pelajari dari modul yang sudah diberikan sebelumnya.

Kamis 11 Juli 2019 (09-23 - 17.17)

Pada hari kedelapan saya melanjutkan project yang sebelumnya, yaitu tentang membuat *CRUD* pada tabel *Courses* yang dimana tabel *courses* ini terhubung dengan tabel *user* dan tabel *studies*, sama halnya dengan tabel *Credits* yang sudah dibuat sebelumnya, langkah pertama yang saya lakukan yaitu yaitu membuat model dan *controller* untuk tabel *Courses* dengan menggunakan syntax `php artisan make:model Course` untuk membuat model dan `php artisan make:controller CourseController -r` untuk membuat *controller*, langkah selanjutnya yaitu membuat deklarasi variabel pada model *Courses* dengan menggunakan *code* `protected $table = 'courses'`, yang dimana data yang diinputkan akan dikirim ke tabel *courses* yang ada di database lalu menambahkan variabel `protected $fillable = ['user_id', 'name', 'credit']` yang dimana *code* tersebut digunakan untuk menyimpan data-data yang akan dikirim ke field-field yang ada di tabel *courses*, sebelum kelangkah selanjutnya disini saya sedikit menjelaskan tentang penggunaan ***protected*** dimaksudkan bahwa variabel yang kita deklarasikan hanya dapat diakses oleh kelas turunan dari *class courses*. Lalu selanjutnya saya menambahkan *code* di fungsi *index* pada file *CourseController* dengan menuliskan *code* `return 'ini data courses'`, setelah itu saya menambahkan route pada file *route.php* dengan menuliskan *code* `Route::get('/courses,`

`'CourseController@index')` yang di mana *code* ini digunakan untuk pemanggilan *url* di *browser*, lalu saya melanjutkan untuk mengerjakan sesuai dengan soal yang diberikan setelah selesai, disini saya tidak ada menemukan kendala, lalu saya melanjutkan mengerjakan *CRUD* pada tabel *periodes* yang dimana pada tabel periode ini hanya berelasi ke tabel *studies*, yaitu tabel master dari program yang akan dibuat.

Jumat 12 Juli 2019 (09.38 - 17.25)

Pada hari kesembilan saya melanjutkan *project* yang sebelumnya, disini saya mulai membuat *CRUD* pada tabel *Periodes* yang dimana tabel *periode* ini berelasi dengan tabel *studies* seperti pada tabel *courses* dan tabel *users*. Langkah pertama yang saya lakukan yaitu membuat model dan *controller* untuk tabel *periodes* dengan menggunakan syntax `php artisan make:model Periode` untuk membuat model dan `php artisan make:controller PeriodeController -r` untuk membuat *controller*, setelah selesai saya melanjutkan untuk membuat deklarasi variabel pada model periode dengan menuliskan *code* `protected $table = 'periodes';` untuk mengirimkan data yang akan dikirim ke tabel *periodes* yang ada di database lalu saya menambahkan variabel `protected $fillable = ['year', 'semester', 'register_at', 'register_end']` yang dimana variabel tersebut digunakan untuk menampung data-data yang sesuai dengan *field-field* yang ada di tabel *periodes* untuk dikirim ke tabel *periodes*. lalu saya menambahkan fungsi *study* yang dimana fungsi ini digunakan untuk merelasikan tabel *periodes* dengan tabel *studies* dengan menuliskan *code* `return $this->hasMany('App\Study');` setelah itu saya melanjutkan untuk menambahkan *code* pada fungsi *index* yang berada di *controller* *periodes* dengan menambahkan *code* `return ('ini view periode');` setelah itu saya membuat routing untuk tabel periode dengan menuliskan *code* `Route::get('/periode', 'PeriodeController@index');` setelah selesai saya mencoba untuk menjalankan programnya dan tidak ada yang error, lalu saya melanjutkan untuk mengerjakan sesuai dengan soal yang diberikan. Setelah selesai

saya menanyakan ke mas Farid tentang tugas yang telah saya kerjakan dan saya menanyakan tentang beberapa hal terutama tentang penggunaan *library Carbon* untuk menentukan tanggal registrasi dan tanggal selesai. Setelah itu mas farid mengecek tentang progres yang saya kerjakan dan saya diminta untuk melanjutkan mengerjakan tabel master yaitu tabel *study*.

Sabtu 13 Juli 2019 (09.28 - 17.06)

Pada hari kesepuluh saya melanjutkan mengerjakan project yang sebelumnya, yaitu tentang membuat *CRUD* pada tabel master yaitu pada tabel *study*. Langkah-langkah yang saya lakukan hampir sama dengan langkah sebelumnya yaitu membuat model dan *controller* dengan menggunakan syntax `php artisan make:model Study` untuk model *study* dan `php artisan make:controller StudyController` untuk *study controller* lalu saya membuat deklarasi variabel dengan menuliskan syntax `protected $table = 'studies'` dan `protected $fillable = ['user_id' , 'courses_id' , 'periode_id', 'grade'];` dan pada model *study* saya menambahkan beberapa fungsi untuk merelasikan tabel *study* dengan tabel-tabel yang dibutuhkan pada soal, disini saya membuat fungsi untuk tabel *courses*, *periodes*, dan *users* dengan menggunakan *code* `return $this->belongsTo('App\Users', 'user_id', 'id');` yang dimana fungsi ini digunakan untuk merelasikan tabel *study* dengan tabel user dengan mengambil id user pada tabel user dan saya mengulangi langkah yang sama untuk fungsi-fungsi selanjutnya. Setelah itu saya membuat routing untuk tabel *study* dengan menuliskan *code* pada file `web.php` `Route::get('/study', 'StudyController@index');`, setelah itu saya mencoba untuk menjalankan programnya dan melanjutkan mengerjakan sesuai dengan soal yang diberikan. Dan disini saya mulai agak bingung tentang soal yang diberikan yang dimana saat mahasiswa registrasi hanya bisa pada periode yang telah diberikan oleh

admin, dan saya mulai menanyakan tentang pertanyaan tersebut kepada mas Farid, dan diminta untuk menggunakan *dateNow()*, dan *package Carbon* pada *Laravel* untuk mengambil tanggal sekarang dan mengedit format tanggal, dan untuk mencari *range* pada periode saat ini sesuai tanggal komputer saya menggunakan *WhereHas*, dan *WhereBetween* pada *Laravel*. Setelah itu saya mulai untuk menjalankan programnya dan melakukan testing sesuai dengan soal yang diberikan dan mas farid melakukan pengecekan tentang tugas yang diberikan dan menanyakan beberapa hal terkait tentang *code* yang sudah saya kerjakan dan pemahaman saya tentang *code* yang saya kerjakan.

Senin 15 Juli 2019 (09.30 - 17.40)

Pada hari kesebelas pukul 10.30 diadakan meeting oleh tim *software* dan saya diajak untuk meeting bersama yang membahas tentang progres yang dikerjakan minggu lalu dan membahas tentang *project* yang akan ditargetkan pada minggu ini, saya diberikan tugas untuk melanjutkan soal pada *project* yang sebelumnya dengan menggunakan ***Laravel-Generator*** yang sudah dibuat oleh tim *software* di JALA, yang dimana karena baru pertama kali saya melihat dan mengetahui tentang *Laravel-Generator*, pada pukul 12.45 saya mulai mempelajari tentang struktur folder dan alur *code* yang ada di *Laravel-Generator*, dan menanyakan beberapa hal tentang *Laravel-Generator* kepada mas Farid. *Laravel-Generator* merupakan sebuah *framework Laravel* yang sudah di *costume* oleh tim *software* JALA yang digunakan untuk membuat sebuah *website* dengan mempermudah kita menuliskan *code* serta memperpendek *code* yang kita buat, pada *Laravel-Generator* ini sudah disediakan semua fungsi *CRUD* pada *Laravel* yang sudah di *costume* sedemikian oleh tim *Software* JALA. Langkah pertama yang saya lakukan yaitu mulai menginstall *Laravel* dengan menggunakan

```
syntax composer create-project --prefer-dist laravel/laravel generator
```

, setelah selesai menginstall *Laravel* saya mulai menginstal *Laravel-Generator* dengan langkah, buka file *Laravel* yang sudah diinstall menggunakan

Visual Studio Code lalu buka terminal dan jalankan `composer update` untuk mengupdate installan yang telah kita lakukan lalu saya mulai menginstal dependensi *Laravel-Generator* dengan menjalankan `composer require atn1c/laravel-generator`, yang dimana setelah kita menjalankan syntax tersebut maka dependensi *Laravel-Generator* sudah otomatis terinstall pada *Laravel* yang sebelumnya sudah kita install, pada struktur folder dan beberapa *code* ada perubahan sesuai dengan dependensi *Laravel-Generator* yang sudah kita install. Setelah itu saya melanjutkan untuk menambahkan tampilan *view* nya dengan menggunakan *template bootstrap AdminLTE* yang sudah dibuat oleh tim *software JALA*, jadi saya tinggal menggunakannya dengan menjalankan `composer require atn1c/laravel-admin-lte` setelah itu saya mencoba untuk menjalankan programnya dan menanyakan ke mas Farid tentang tugas yang telah saya kerjakan.

Selasa 16 Juli 2019 (09.30 - 17.40)

Pada hari kedua belas saya mulai mengerjakan *CRUD* pada tabel *credit* di *Laravel-Generator*, strukturnya hampir sama dengan *Laravel* biasa tetapi disini kita tidak perlu membuat tampilan *view*-nya, karena sudah ke *generate* dengan tampilan *bootstrap AdminLTE* langkah pertama yang saya lakukan yaitu membuat *migration*, model dan *controller* dengan menggunakan `php artisan make:controller --model=Credit CreditController`, setelah menjalankan syntax tersebut maka otomatis semua file dan folder yang diperlukan untuk *CRUD credit* sudah otomatis dibuat dan juga tampilan *view* nya juga sudah otomatis terbuat jadi disini saya menambahkan beberapa *code* untuk *migration* sesuai dengan soal yang diberikan, lalu menambahkan fungsi pada model yaitu fungsi *users* yang digunakan untuk merelasikan tabel *credit* dengan user lalu saya membuat *form* input untuk tabel *credit* pada controller *credit*, pada *Laravel-Generator* kita tidak membuat *view*-nya karena sudah otomatis ter-*generate* dengan dependensi *AdminLTE* yang sudah kita install sebelumnya, untuk membuat *view*-nya dengan menuliskan *code* pada fungsi *fields* [

'field' => 'input', 'type' => 'text', 'name' => 'credit', 'label' => ucwords(__('Credit in this semester')), 'required' => true], *code* tersebut digunakan untuk membuat sebuah *form* input pada tabel credit, lalu saya melanjutkan untuk membuat *code* untuk menambahkan data kedalam tabel credit dengan menambahkan *code* `return ['store' => ['user_id' => 'required|integer', 'credit' => 'required|integer|max:24',],].` Setelah itu saya mencoba untuk menjalankan programnya dan saya menanyakan hal yang saya kerjakan kepada mas Farid, dan sudah sesuai, lalu saya lanjutkan untuk membuat *role login user* dengan menggunakan autentifikasi *Policy* pada *Laravel-Generator*, yang dimana pada tabel Credit mahasiswa hanya diberikan hak akses untuk melihat saja.

Rabu 17 Juli 2019 (09.20 - 17.30)

Pada hari ketiga belas saya melanjutkan *project* yang sebelumnya, dimana sekarang saya mulai mengerjakan *CRUD* pada tabel yang lainya langkahnya hampir sama dengan langkah sebelumnya yaitu membuat sebuah model dan *controller* dengan menggunakan `php artisan make:controller --model=Course CourseController`. Setelah itu saya melanjutkan untuk menambahkan *code-code* yang diperlukan sesuai dengan langkah sebelumnya. Lalu saya mengerjakan *Policy* untuk hak akses ke tabel pada setiap tabel, pada folder *policy* sudah tersedia file model *policy* untuk mempermudah kita membuat *policy* pada masing-masing tabel langkah pertama yang saya lakukan yaitu membuat sebuah fungsi *create* pada file `creditpolicy.php` yang terletak pada folder ***Policy*** dengan menuliskan *code* `$allow = $user->role == 'admin'; return $allow;;` yang dimana *code* tersebut digunakan untuk memberikan hak akses kepada admin untuk mengakses fitur *create* data pada tabel credit, dan saya melakukan langkah yang sama pada tabel-tabel selanjutnya, dan menyesuaikan dengan soal yang diberikan. Setelah selesai saya menanyakan ke mas Farid, tentang penggunaan *policy* yang saya buat masih ada yang salah yaitu pada tabel *study*, yang dimana penulisan *plural* pada *Laravel-Generator* yaitu menggunakan *study*, bukan *studi*, setelah saya *solved error* nya, saya

menanyakan ke mas Farid, dan sudah selesai, lalu saya melanjutkan untuk mengerjakan fungsi yang lainnya.

Kamis 18 Juli 2019 (09.28 - 17.06)

Pada hari ke empat belas saya melanjutkan mengerjakan *project* yang sebelumnya yang dimana pada hari ini saya mengerjakan tentang penentuan tanggal registrasi kepada mahasiswa, yang dimana tanggal akan di set oleh admin pada tabel periode, dan mahasiswa hanya bisa menginput dan mengedit data pada tabel *study* sesuai dengan periode yang telah ditentukan, disini saya menggunakan fungsi *dateNow()*, sama dengan pada *Laravel* sebelumnya, tetapi disini kita tinggal memanggil tanggal komputer saja, untuk *query* nya sudah tersedia pada *Laravel* generator. Langkah pertama yang saya lakukan yaitu membuat sebuah fungsi pada file *PolicyStudi.php* pada folder *policy* untuk mengambil data tanggal di komputer dan membandingkannya dengan data pada field *register_at* dan *register_end* pada tabel periode dengan menuliskan code `$allow3 = $model->periode()->whereDate('register_at', '<=', now())->whereDate('register_end', '>=', now())->exists();` disini saya membuat variabel penampung dengan nama *\$allow3* untuk menampung data tanggal yang akan dibandingkan lalu saya menggunakan fungsi *whereDate()* untuk mencari *range* nilainya dengan membandingkan jika tanggal *register_at* lebih kecil sama dengan tanggal sekarang(tanggal komputer) maka akan dibandingkan dengan tanggal *register_end* lebih besar sama dengan tanggal sekarang maka akan menampilkan data yang bisa diinputkan oleh *role* yang telah ditentukan, untuk menampilkan datanya saya membuat code `return $allow1 or ($allow2 and allow3)`, yang dimana untuk variabel *\$allow1* dan *\$allow2* menampung nama *role* yang akan diijinkan. Setelah selesai mas Farid melakukan pemeriksaan dan saya menanyakan tentang alur program untuk menentukan tanggal periode yang hanya boleh menginputkan data. Ketika sudah selesai dan saya diminta untuk menunggu tugas selanjutnya.

Senin 22 Juli 2019 (09.20 – 17.00)

Pada hari kelima belas, seperti biasa tim *software* JALA mengadakan rapat mingguan untuk membahas hal apa yang akan dikerjakan pada minggu ini, saya mempresentasikan tugas yang telah saya buat kepada seluruh tim *software* JALA, dan ada beberapa pertanyaan yang diberikan seputaran tugas yang saya kerjakan setelah itu saya mempelajari lagi tentang tugas yang saya kerjakan dan mempelajari seputaran struktur *code* dan struktur folder pada *Laravel-Generator* dan setelah itu saya belum diberikan tugas dikarenakan tim *software* JALA masih menentukan fitur baru yang akan dirilis. Sembari menunggu tugas selanjutnya mas Farid meminta saya untuk mengerjakan *CRUD* user pada *project* sebelumnya agar *role* sebagai admin bisa menambahkan user dan panel admin, langkah pertama yang saya lakukan yaitu membuat *controller* dan model user yang dimana model dengan menggunakan syntax `php artisan make:model --model=User UserController`, setelah itu saya menghapus beberapa file user yang ter-*generate* dikarenakan ada duplikasi file user yang saya buat dengan file user bawaan dari *Laravel*, disini saya menambahkan fungsi untuk input pada *controller* user dengan langkah yang sama pada sebelumnya dan untuk langkah selanjutnya saya membuat sebuah routing untuk user yang berbeda yang dimana sebelumnya saya sudah menginstall autentifikasi *login* pada *Laravel* dan disini saya mulai menambahkan routing user pada file `web.php` dengan menuliskan syntax `Route::get('/user', 'UserController@index');` untuk mengambil semua data user yang di database agar ditampilkan di page admin, dan membuat fungsi *CRUD* selanjutnya sesuai dengan langkah yang sudah saya buat sebelumnya.

Selasa 23 Juli 2019 (09.35 – 17.40)

Pada hari keenam belas saya diminta untuk membantu tim *software* untuk memasang *SSD* pada laptop, setelah selesai, saya belum diberikan tugas, tapi saya mencoba lagi belajar-belajar tentang menggunakan *Laravel-Generator*. Dan mencoba untuk menjalankan dan melakukan testing tugas yang sebelumnya yang sudah saya

kerjakan, setelah itu saya diminta oleh mas Farid untuk belajar tentang *Git*, *Github*, *Gitlab* dan *Gitflow*, karena sebelumnya saya sudah pernah menggunakan *git* jadi disini saya hanya mengulangi dan memperdalam untuk mempelajari tentang *git*, sebelum kelangkah selanjutnya disini sedikit saya jelaskan apa itu *git*, ***git*** adalah salah satu sistem pengontrol *version* (*Versions Control Sistem*) pada proyek perangkat lunak yang diciptakan oleh **Linus Torvads** pengontrol *version* bertugas untuk mencatat semua perubahan yang dilakukan sendiri maupun banyak orang disini *git* adalah *version control* nya dan untuk *Github* dan *Gitlab* adalah sebuah sistem *cloud* untuk menyimpan semua perubahan *code* yang kita buat, yang dimana *github* dan *gitlab* sudah terkoneksi langsung dengan *git*. Langkah pertama yang saya lakukan yaitu membuat sebuah *repository* di *Github* dan dikarenakan saya sudah memiliki akun *github* jadi disini saya langsung saja untuk membuat *repository* nya, setelah itu saya menghubungkan *github* dengan *git* yang sudah saya install di komputer saya, untuk menghubungkannya dengan menggunakan syntax `git config --global dewa.g.ari28@gmail.com` yang dimana disini saya menghubungkan *git* dengan *github* dengan menggunakan nama *email* yang saya gunakan saat register di *github*, setelah itu saya melanjutkan untuk membuat *remote git* dengan *github* dengan menggunakan syntax `git remote add origin https://github.com/cecepaf18/Tugas-2.git` yang dimana pada langkah ini saya sudah bisa meremote *repository github* yang saya buat dengan *git*, setelah itu saya melanjutkan untuk mempelajarinya dengan mengikuti tutorial yang saya pelajari di internet.

Rabu 24 Juli 2019 (09.28 – 17.06)

Pada hari ke tujuh belas saya dimasukan ke *GitLab* tim *Software* JALA, yang dimana saya diminta untuk belajar tentang penggunaan *Git* terlebih dahulu, karena sebelumnya saya juga sudah pernah belajar menggunakan *Git* pada Mata Kuliah P3L pada semester 6, dan beberapa *project* yang saya buat juga saya simpan di *Git*, dan

pada hari sebelumnya saya juga sudah pernah mempelajari tentang *Git*, jadi selanjutnya saya meng *clone* file *WebApp JALA* pada *github*, dengan menggunakan syntax `git remote add origin https://gitlab.com/atnic/jala-web.git` yang dimana saya meng *cloning* program *WebApp JALA* dari *github* dengan menggunakan syntax tersebut, setelah itu saya mempelajari alur *code* nya terlebih dahulu, sebelum diberikan tugas langsung. Untuk alur *code WebApp JALA* sama seperti *Laravel-Generator* yang sudah saya pelajari sebelumnya dikarenakan pada *WebApp JALA* ada beberapa *code* yang ditambahkan dan menggunakan banyak *controller* model dan tabel jadi disini saya menanyakan beberapa hal tentang struktur *code* yang ada kepada mas Farid. Dan mas Farid menjelaskan tentang struktur *code* nya, tetapi disini saya belum diberikan untuk menjalankan programnya langsung dikarenakan saya harus memahami dan mempelajari struktur *code* nya terlebih dahulu, pertama saya diminta untuk memahami tentang alur *code* pada data **Product** yang dimana data produk memiliki beberapa relasi ke tabel yang lainnya, dan mempelajari tentang fungsi-fungsi yang ada pada data *product* setelah itu saya menanyakan tentang beberapa fungsi yang ada pada data *product* kepada mas Farid, seperti fungsi *fields* yang digunakan untuk menginputkan data pada *form* dan pada fungsi *fields* saya menanyakan tentang *code* `$has_purchase_items = $product && $product->purchase_items()->exists();` yang dimana fungsi tersebut digunakan untuk memanggil fungsi *purchase_item* yang berada di *controller purchase_item* yang digunakan untuk menampilkan data harga *product* per itemnya.

Kamis 25 Juli 2019 (09.28 - 17.06)

Pada hari ke delapan belas saya masih melanjutkan tugas yang sebelumnya yaitu tentang mempelajari *code* program pada aplikasi *JALA*, pada aplikasi *WebApp JALA* menggunakan *Laravel-Generator*, dan *front-end* nya menggunakan *Vue.js*, jadi karena saya baru mengetahui tentang *Vue.js*, saya juga menanyakan dan berbincang-bincang dengan mas Dama selaku *Front-End engineer JALA*, dan merupakan alumni

Atmajaya juga, jadi saya lebih memahami ketika mas Dama menjelaskan tentang *vue.js*. *Vue.js* merupakan sebuah *framework* untuk *front-end* yang dikembangkan dengan kerangka kerja **JavaScript** progresif yang bersumber terbuka untuk membangun antarmuka pengguna. Integrasi ke dalam proyek yang menggunakan pusaka *JavaScript* lainnya mudah dilakukan dengan *Vue.js* karena dirancang untuk diadopsi secara bertahap, setelah itu saya menanyakan ke mas Dama tentang penulisan *code* di *Vue.js*, sebelum kelangkah tersebut saya diminta untuk menginstall *Vue.js* terlebih dahulu dengan menggunakan syntax `npm install vue`, dikarenakan saya sudah menginstall *package npm* dan *node.js* sebelumnya jadi saya cukup menjalankan syntax tersebut untuk menginstall *vue.js*, setelah itu saya menanyakan ke mas Dama tentang struktur *code* dan folder pada *Vue.js*, lalu saya mencoba membuat sebuah tampilan pada *vue.js* dengan membuat file `create.vue` pada folder *component* yang berada di *vue.js*, lalu saya mengisi file tersebut dengan *code* `<template> This will never change: {{ msg }}` yang dimana *code* tersebut digunakan untuk menampilkan text pada tampilan web, lalu saya membuat routing pada *Vue.js* dengan menuliskan *code* `component('createt',require('./ components/ create.vue'). default);` yang dimana *code* tersebut digunakan untuk membuat routing pada *Vue.js* di file *app.js*, lalu saya mencoba untuk menjalankan programnya dan melanjutkan membuat tampilan *vue.js* yang selanjutnya.

Jumat 26 Juli 2019 (09.20 - 17.30)

Pada hari ke sembilan belas, seperti biasa diadakan rapat akhir minggu yang membahas tentang progres yang telah dikerjakan, dan saya sudah diijinkan untuk mulai menjalankan programnya, dengan terlebih dahulu membuat database pada *phpmyadmin*, dan menjalankan *migration* pada *Laravel-Generator* dengan menggunakan syntax `php artisan migrate`, yang dimana tabel-tabel yang diperlukan sudah langsung tersedia ketika saya meng *cloning* program *WebApp JALA*, jadi saya tinggal

menjalankan syntax tersebut untuk menyambungkan tabel-tabel yang ada dengan database yang telah saya buat sebelumnya, dan setelah menjalankan syntax tersebut disini saya sempat menemukan *error*, yang dimana data *dummy* yang dibuat tidak ikut ter-generate pada saat saya menjalankan *migration*, maka dari itu saya harus menjalankan *php artisan seeder* pada *Laravel* agar data *dummy* nya otomatis terisi. Setelah itu saya mencoba untuk menjalankan programnya dengan melakukan register terlebih dahulu untuk membuat akun dan mencoba untuk *logIn*. Pada saat *logIn* saya hanya masuk sebagai user dengan *role* pelanggan bukan admin, jadi disini saya mengganti *role* nya dengan admin agar saya bisa mengakses semua fitur yang ada di *WebApp JALA*, untuk mengganti *role* nya saya masuk ke database yang sudah saya buat lalu mencari tabel user dan mencari akun yang saya gunakan untuk register, disini saya mengubah *role* saya menjadi admin, yang dimana pada program *WebApp JALA* hanya ada 2 *role* yaitu admin dan pelanggan, setelah itu saya mempelajari alur programnya dan beberapa struktur *code* yang mengacu pada fitur-fitur yang ada di *WebApp JALA*, setelah itu saya menanyakan beberapa *code* dan alur jalanya program kepada mas Farid, dan mas Farid memberikan penjelasan tentang alur programnya.

Selasa 30 Juli 2019 (09.35 – 17.40)

Pada hari ke duapuluh, saya mulai menjalankan programnya dan mulai membaca *code* program yang diberikan, disini saya masih mempelajari tentang *code* program yang diberikan dan mempelajari tentang struktur *code* pada *Laravel-Generator* sesuai dengan penjelasan mas Farid pada hari sebelumnya, saya mempelajari alur kerja program khususnya pada fitur buat tambak, fitur ini digunakan oleh pelanggan untuk membuat simulasi tambak udang, langkah pertama yang saya lakukan yaitu membuat sebuah tambak dengan nama Tambak Jogja, lalu mengisi *form* alamat dan mengisi jumlah kolam yang diinginkan, setelah itu akan muncul jumlah kolam yang sudah kita buat, disini saya membuat 5 kolam, setelah itu akan muncul informasi kolam yang dimana disini kita bisa mengubah nama kolam dan mengisi luas kolam, setelah

itu langkah yang terakhir melakukan verifikasi dengan kolam yang sudah kita buat, setelah itu akan muncul tampilan lihat tambak, disini kita bisa melihat perkembangan tambak kita dan mengisi jumlah udang yang akan dimasukan ke setiap tambak, setelah itu saya melanjutkan untuk mempelajari alur program pada fitur-fitur yang lainnya sembari mempelajari setiap alur *code* pada fitur-fitur tersebut, setelah selesai saya menanyakan ke mas Farid tentang pemahaman yang saya sudah kerjakan, lalu saya diminta untuk belajar tentang *Git flow* untuk mengerjakan *project* yang akan diberikan pada hari selanjutnya.

Rabu 31 Juli 2019 (09.28 - 17.06)

Pada hari ke duapuluh satu, saya mulai dibuatkan satu *branch* pada *GitLab* yaitu *branch Laravel-Generator-0.10* yang dibuatkan oleh mas Farid disini saya mulai meng *clone* ulang program *WebApps* master dari JALA, dan saya mempelajari tentang penggunaan *Git* secara lanjut, setelah itu saya diberikan tugas untuk membuat *update* setiap fitur yang ada di *WebApps* JALA, dengan menggunakan *Laravel-Generator-0.10* yang sudah dibuat oleh tim *software* JALA. Sebelum saya melakukan *update* saya diminta untuk mempelajari penggunaan *git*, disini saya diminta untuk menggunakan *git bash*, atau *git* dengan syntax jadi cukup menjalankan beberapa syntax sesuai dengan panduan di *ebook git* yang sudah saya pelajari, pada *git* ada beberapa syntax yang saya pelajari diantaranya *git status*, yang dimana syntax ini digunakan untuk melihat status *update-an code* pada file yang telah kita lakukan, lalu ada *git add .* yang dimana syntax ini digunakan untuk merubah status file agar tersimpan pada *staging*, tanda *'.'* yang artinya kita menyimpan semua perubahan pada file yang telah kita *update* ke *staging*, lalu saya mempelajari tentang penggunaan syntax *git commit* syntax ini digunakan untuk merubah semua

penyimpan file menjadi *committed* agar semua perubahan disimpan oleh *git*. Setelah itu saya mengecek semua perubahan yang telah disimpan dengan menggunakan syntax `git status`, untuk mengetahui perubahan apa saja yang telah disimpan oleh *git*, setelah itu saya mempelajari lebih lanjut tentang penggunaan syntax pada *git*, dan menanyakan tentang pembuatan *branch*, *git hotfix* dan lainnya kepada mas Farid, dan juga saya mempelajari tentang penulisan syntax *git* pada *ebook* yang saya download dari Website *git* nya langsung.

Kamis 1 Agustus 2019 (09.37 – 17.26)

Pada hari kedua puluh dua, saya melanjutkan untuk memahami struktur programnya dan memahami alur fitur-fitur yang berada di *WebApps JALA*, setelah itu saya menanyakan beberapa hal kepada mas Farid tentang fitur mana saja yang di *update* dan *update*-nya seperti apa, setelah mendengar penjelasan mas Farid, saya mulai meng-*update* satu fitur yaitu fitur *Parameters* dengan terlebih dahulu menjalankan *Composer update* dan ***composer require atnic/laravel-generator*** pada terminal *VsCode* yang dimana syntax tersebut digunakan untuk meng *clone* dan menggenerate file *Laravel-Generator* dari *github* *Atnic* yang sudah disediakan oleh tim *software JALA*, lalu saya menghapus beberapa file yang berkaitan dengan *Parameters*, yaitu *File Factories*, *Models*, *Filters*, *Policy*, *lang*, *view*, *Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:controller --model=Parameter ParameterController`, yang dimana syntax tersebut digunakan untuk membuat file-file pada Fitur *Parameter* yang telah dihapus. Setelah itu saya mulai mengerjakan *update*-an pada fitur *parameters* langkah pertama yang saya lakukan yaitu mengubah *code* pada fungsi *visibles* yang sebelumnya *code* nya berupa ['name' => 'product_category', 'label' => 'products.product_category'), 'column' => 'name_translated'], saya rubah menjadi ['name' => 'product_category', 'label' => ucwords(__('products.product_category')), 'column' => 'name_translated'], yang dimana *update*-an tersebut sudah sesuai dengan

penulisan *code* pada **Laravel generator 0.10** yang sudah dibuat oleh tim *software* JALA. Selanjutnya saya melakukan update pada fitur-fitur yang telah di hapus dan menyesuaikan penulisan *code* dengan *Laravel generator 0.10*, setelah itu saya mencoba untuk menjalankan programnya, dan saya menemukan beberapa error, dan saya mencoba untuk mencari dan *men-solved* error tersebut.

Jumat 2 Agustus 2019 (09.40 - 17.45)

Pada hari kedua puluh tiga saya melanjutkan untuk *men-solved code* yang *error* pada fitur *parameters* setelah menemukan *error*-nya yaitu disebabkan karena salah penulisan saat pembuatan modelnya, jadi tidak *ter-generate* secara otomatis, dan saya mengulang langkah langkah pembuatan fitur *parameters* seperti hari sebelumnya, setelah itu saya melanjutkan untuk melakukan *update* pada *code* program di beberapa file *Parameters*, disini saya melakukan *update* pada file `parameterfilter.php` yaitu membuat sebuah variabel `$sortables` dengan berbentuk *array*, yang dimana didalamnya berisi *field-field* yang berada pada *view index* parameter yang bisa di sorting secara *ascending* dan saya juga membuat variabel `$searchable` yang berbentuk *array* yang dimana didalamnya berisi *field-field* yang berada pada *view index* parameter yang bisa di *search* berdasarkan *field-field* yang akan dicari, setelah itu saya menghapus semua fungsi yang berada pada file `ParameterPolicy.php`, dikarenakan fungsi-fungsi tersebut sudah di *handle* oleh file `modelpolicy.php` jadi disini saya tinggal menambahkan *code* `Permission::generateFor('parameter');` yang dimana untuk fitur parameter hanya bisa diakses oleh *role* yang bertipe admin yang sudah di *handle* oleh fungsi yang berada di file `modelpolicy.php`, setelah itu saya mencoba untuk menjalankan programnya dan melakukan pengetesan untuk fitur parameter yang telah di *update*, dan saya juga menanyakan beberapa hal tentang penulisan *update*-an yang saya lakukan ke mas Farid, setelah selesai sama melakukan *git commit* ke *gitlab* dengan *branch* *Laravel-generator.0.10* yang sudah dibuatkan sebelumnya.

Senin 5 Agustus 2019 (09.45 – 17.30)

Pada hari kedua puluh empat, seperti biasa saya mengikuti rapat tim *software* JALA, tentang progres apa yang sudah dikerjakan pada minggu sebelumnya dan melanjutkan progres yang akan dicapai pada minggu ini, setelah selesai rapat saya bertanya kepada mas Farid tentang progres *update* fitur *parameters* yang telah saya kerjakan, dan mas Farid mengatakan sudah sesuai, lalu saya diminta untuk melanjutkan untuk melakukan *update* fitur-fitur yang lainnya yang berada pada data master. Fitur yang saya *update* selanjutnya yaitu fitur *product* langkah yang saya lakukan hampir sama dengan langkah sebelumnya yaitu menghapus beberapa file yang berkaitan dengan fitur *product*, yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:model --model=Product ProductController`, setelah itu saya mengecek apakah sudah *ter-generate* semuanya atau belum, lalu saya melanjutkan untuk melakukan *update* pada file *ProductController* disini saya akan jelaskan untuk *update* pada fungsi *fields* yang dimana fungsi ini digunakan untuk memasukan data yang akan diinputkan ke *form* yang ditampilkan pada *view*, langkah pertama yang saya lakukan yaitu mengecek *code* yang di *update* dan yang tidak disini saya melakukan *update form* inputan untuk *name* dan *unit*, dikarana hanya 2 *form* tersebut yang belum sesuai, untuk meng-*update*-nya saya menuliskan *code* `['field' => 'input', 'type' => 'text', 'name' => 'name', 'label' => ucwords(__('products.name')), 'required' => true],` *code* tersebut sudah menyesuaikan dengan penulisan *update-an code* pada *Laravel-Generator*, lalu saya melakukan langkah yang sama dengan langkah sebelumnya, setelah itu saya melakukan pengetestan fitur *product* yang telah saya *update* dan menanyakan ke mas Farid tentang progres yang telah saya kerjakan.

Selasa 6 Agustus 2019 (09.40 – 17.35)

Pada hari kedua puluh lima, saya melanjutkan untuk melakukan *update* fitur *Devices* dengan langkah langkahnya sesuai dengan hari sebelumnya, yaitu langkah pertama yang saya lakukan menghapus beberapa file yang berkaitan dengan fitur *Device* sesuai dengan *Laravel-Generator*, file yang saya hapus yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya melakukan *update* pada file *DeviceController* yang dimana disini saya menjelaskan *update* pada fungsi *rules* dengan menambahkan code 'store' => ['imei' => 'required|string|max:255|unique:devices', 'name' => 'string|max:255|nullable', 'firmware' => 'string|max:255|nullable',], yang dimana *code* tersebut digunakan untuk memvalidasi *form* input yang kita lakukan dan mengirimkan data yang kita inputkan ke dalam database, setelah itu saya melakukan *update* pada folder *view* untuk fitur *device* disini saya hanya mengubah ukuran tampilan data table pada *view index* yang awalnya 8 menjadi 12 dengan menggunakan code 'col_class' => 'col-md-12', yang dimana *code* tersebut hanya mengubah ukuran tampilan pada data table fitur *Device*, lalu saya melakukan pengetestan sistem untuk menyesuaikan fitur *Device* yang telah saya *update* dengan *Laravel-Generator*, setelah itu saya menanyakan ke mas Farid tentang progres yang telah saya kerjakan dan sudah sesuai, lalu diminta untuk melakukan *commit* ke *gitlab* dengan perubahan *code* yang telah saya kerjakan karena pada fitur-fitur yang saya *update* sebelumnya ada beberapa yang belum di *commit*, jadi disini saya menjalankan syntax `git status` untuk mengecek fitur mana saja yang belum saya *commit*, lalu saya menjalankan syntax `git add .` untuk menyimpan semua perubahan yang telah saya lakukan ke dalam *staging*, setelah itu saya menjalankan syntax `git commit -m "Update fitur Parameter, Product dan Device"`, yang dimana syntax *-m* digunakan untuk memberikan pesan bahwasanya fitur-fitur tersebut telah saya *update*.

Rabu 7 Agustus 2019 (09.25 – 17.40)

Pada hari kedua puluh enam saya melanjutkan mengerjakan *update* pada fitur *Devices Parameters* yang dimana langkahnya sama dengan langkah sebelumnya, sesuai dengan *Laravel-Generator*, file yang saya hapus yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:model --model=DeviceParameter DeviceParameterController`, setelah itu saya melakukan *update* pada file *DeviceParameterFactory* yang dimana disini saya menambahkan syntax `use App\DeviceParameter` untuk memanggil *library deviceParameter*, lalu saya menambahkan fungsi `$factory-> define(App\DeviceParameter:: class, function (Faker $faker) yang dimana me-return data yang ada di tabel deviceParameter dengan menuliskan code return ['code' => $faker->userName,]` yang dimana *code* ini digunakan untuk menginputkan data *dummy* ke dalam tabel *deviceParameter*, setelah selesai saya melanjutkan untuk menjalankan programnya dan disini saya menemukan *error* pada model *Policy* nya yang dimana ada *update* terbaru pada *Laravel-Generator* yang dibuat oleh mas Farid jadi saya harus menjalankan *cloning* pada *Laravel-Generator* dengan menjalankan syntax `composer require atnrc/laravel-generator`, lalu menjalankan *Composer Update*, lalu saya mencoba menanyakan *update* yang dilakukan oleh mas Farid pada model *Policy* nya dan disini mas Farid menambahkan satu fungsi yang bernama `getModel()` yang digunakan untuk *menghandle role login* yang diperbolehkan untuk mengakses fitur-fitur yang ditentukan, lalu saya mencoba untuk mempelajari tentang alur *code* nya dan menanyakan ke mas Farid tentang fungsi tersebut. Setelah itu saya mencoba untuk menjalankan programnya dan mengecek *update-an* pada fitur *deviceParameter* yang telah saya lakukan, lalu saya menanyakan ke mas Farid tentang progres yang telah saya kerjakan. Dan mas Farid meminta saya untuk melanjutkan *update-an* pada fitur selanjutnya.

Kamis 8 Agustus 2019 (09.30 – 17.30)

Pada hari kedua puluh tujuh saya melanjutkan untuk melakukan *update* pada fitur, *Products Categories* dengan langkah yang sama pada hari sebelumnya, sesuai dengan *Laravel-Generator*, file yang saya hapus yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:model --model=ProductCategory ProductCategoryController`, setelah itu saya melakukan *update* pada file `ProductCategoryControllerTest.php` pada file ini saya hanya menambahkan *code* namespace `Test/Feature` dan `use App\ProductCategory` yang dimana *code* tersebut digunakan untuk memanggil model *ProductCategory*, untuk fungsi-fungsi yang lainya tidak ada perubahan dikarenakan sudah sesuai dengan *code* sebelumnya, sebelumnya saya menjalankan programnya saya mengecek *code* nya terlebih dahulu karena ada *update* terbaru di model *Policy* nya maka saya harus menambahkan *permission* pada *seeder*, dengan mengetikan `Permission::generate('product_category')`, yang digunakan untuk memberikan akses *Permission* kepada user. Setelah itu saya mencoba untuk menjalankan programnya dan mengecek apakah sudah sesuai dengan *update*-an yang saya lakukan, setelah itu saya menanyakan ke mas Farid tentang progres yang telah saya kerjakan, dan mas Farid mengatakan sudah sesuai, sebelum saya lanjut ke tahap berikutnya saya juga menanyakan tentang *code* yang sudah saya kerjakan dan menanyakan tentang *Permission* yang digunakan untuk memberikan hak akses ke user, disini mas Farid menjalankan untuk penggunaan *permission* disini untuk mempersingkat penulisan *code* pada file *policy* pada masing-masing fitur karena sudah di *handle* oleh fungsi `getModel()` yang berada pada file *ModelPolicy*, selain mempersingkat penulisan *code*, juga mempermudah tim *software JALA* untuk melakukan *maintenance* ketika ada kesalahan yang dilakukan.

Jumat 9 Agustus 2019 (09.28 - 17.36)

Pada hari kedua puluh delapan saya melanjutkan untuk melakukan *update* pada fitur *Tags, Species* dan *Inventarisasi*, langkahnya sama dengan hari sebelumnya, yang

dimana pada proses ini saya melakukan pengecekan pada *code* dikarenakan pada fitur *Tags* dan *Species* berelasi dengan data *Credintials* untuk mengatasi hal tersebut saya menambahkan relasi model nya *Tags* dan *Species* dengan menuliskan *code* `use Filterable` dan menambahkan deklarasi variabel dengan menuliskan *code* `protected $filters = 'App\Filters\TagFilter'` untuk relasi nya saya menambahkan fungsi `public function farm(){return $this->belongsTo(Farm::class);}` untuk merelasikan tabel *Tags* dengan *Farm*, dan saya melakukan langkah yang sama pada fitur *Species* dengan menambahkan *code* pada Model *Species* `protected $filters = 'App\Filters\TagFilter'` dan untuk merelasikan tabel nya saya menambahkan fungsi `public function farm(){return $this->belongsTo(Farm::class);}` untuk merelasikan tabel *species* dengan tabel *farm*. Setelah selesai saya melakukan pengecekan dan pengetesan pada semua fitur yang telah di *update*, lalu menanyakan kepada mas Farid tentang progres saya, yang telah saya lakukan. Lalu mas Farid meminta saya untuk untuk meng *push* pekerjaan saya ke *gitlab*, dikarenakan beberapa fitur yang telah saya *update* belum saya *push* ke *gitlab* maka dari itu saya melakukan pengecekan terhadap semua *code* yang telah saya buat, lalu diminta untuk melakukan *commit* ke *gitlab* dengan perubahan *code* yang telah saya kerjakan, karena pada fitur-fitur yang saya *update* sebelumnya ada beberapa yang belum di *commit*, jadi disini saya menjalankan `git status` untuk mengecek fitur mana saja yang belum saya *commit*, lalu saya menjalankan `git add .` untuk menyimpan semua perubahan yang telah saya lakukan ke dalam *staging*, setelah itu saya menjalankan `git commit -m "Update fitur Tags, Productkategori dan Deviceparameter"`, yang dimana `git commit -m` digunakan untuk memberikan pesan bahwasanya fitur-fitur tersebut telah saya *update*.

Senin 12 Agustus 2019 (09.40 – 17.35)

Pada hari kedua puluh sembilan seperti biasa saya mengikuti rapat tim *software* JALA untuk membahas tentang hal yang saya kerjakan pada minggu lalu dan hal yang akan dikerjakan untuk minggu ini, dan minggu ini saya mendapatkan tugas untuk melanjutkan *update*, kali ini saya melakukan *update* pada data *Credentials* yaitu pada fitur *investors*, langkahnya sama dengan hari sebelumnya, sesuai dengan *Laravel-Generator*, file yang saya hapus yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan `php artisan make:model --model=Investor InvestorController`, lalu saya melakukan *update* pada file *controller* investor dengan menambahkan *code* pada fungsi *relations* yang digunakan untuk merelasikan tabel *investors* dengan *investorProject* untuk membuat fungsi tersebut saya menuliskan *code* `'investor' => ['belongsToMany' => [], 'hasMany' => [['name' => 'investor_users', 'label' => ucwords(__('investors.investor_users'))]`, yang dimana *code* tersebut digunakan untuk merelasikan tabel investor dengan investorproject dengan menggunakan relasi ***belongsToMany***, yang dimana semua data yang ada pada tabel investor juga dimiliki oleh tabel InvestorProject, lalu saya melakukan *update* pada fungsi-fungsi selanjutnya sesuai dengan progres yang saya kerjakan sebelumnya, setelah selesai saya melakukan pengecekan *code* yang sudah dikerjakan, dan menanyakan kepada mas Farid tentang hasil yang sudah saya kerjakan. Dan saya juga menanyakan tentang relasi *belongsToMany* yang dibuat, dan pada *Laravel-Generator*, karena saya baru pertama kali melihat penulisan relasi seperti itu di *Laravel*, dan mas Farid menjelaskan, dikarenakan kita sudah menggunakan *Laravel-Generator* yang dimana semua fungsi sudah di custom sedemikian rupa, jadi kita hanya cukup memanggil fungsi tersebut dengan menuliskan *code* seperti itu.

Selasa 13 Agustus 2019 (09.30 – 17.45)

Pada hari ketiga puluh saya melanjutkan untuk melakukan *update* pada fitur akses dan hak akses, langkahnya hampir sama dengan hari sebelumnya, sesuai dengan

Laravel-Generator, file yang saya hapus yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:model --model=Feature FeatureController`, setelah itu saya melakukan *update* pada beberapa *code* sesuai dengan langkah sebelumnya, tetapi disini saya menambahkan beberapa *permission* pada class *PermissionsTableSeeder* untuk memberikan hak akses ke setiap user, disini saya menambahkan beberapa *code* dengan menuliskan *code* `Permission::generateFor('regions');` `Permission::generateFor('species');` `Permission::generateFor('parameters');` `Permission::generateFor('rfids');` `Permission::generateFor('tags');` *code* tersebut digunakan untuk memberikan hak akses kepada beberapa user untuk mengakses fitur-fitur yang diberikan, setelah selesai saya menanyakan ke mas Farid tentang sistem hak Akses user yang digunakan oleh JALA, dan di JALA menggunakan sistem *Oauth* yang berada di *Laravel* untuk akses user-nya. Dan saya menanyakan tentang sistem *Oauth* pada *Laravel* yang mana saya belum pernah menggunakan sistem tersebut, disini mas Farid menjelaskan tentang penggunaan sistem *Oauth*, *Oauth* merupakan sebuah *library Passport* yang digunakan *Laravel* untuk membuat sistem keamanan yang dimana *library* ini beda dengan *library JWT* yang hanya bisa memberikan hak akses ke user yang telah ditentukan, tetapi pada sistem *Oauth* memberikan *filtering* untuk user ketika mengakses beberapa fitur, untuk langkah install *library Oauth* pada *Laravel* dengan menggunakan syntax `$composer require laravel/passport`, setelah itu kita bisa mengecek apakah dependensi nya sudah terinstal, dengan mengakses file `config.php` dan melihat pada baris terakhir `Laravel\Passport\PassportServiceProvider::class`, yang berarti *library Oauth* sudah berhasil diinstal.

Kamis 15 Agustus 2019 (09.45 – 17.15)

Pada hari ketiga puluh satu saya melanjutkan untuk melakukan *update* pada fitur Peran/*Roles*, langkahnya hampir sama dengan hari sebelumnya, sesuai dengan *Laravel-Generator*, file yang saya hapus yaitu *File Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:model --model=Role RoleController`, lalu saya melakukan *update* sesuai dengan langkah pada progres sebelumnya, lalu saya mencoba untuk menjalankan programnya dan membuat satu *role* baru untuk user, disini saya membuat *role* dengan nama Dewa, dan *display name* dengan nama Mahasiswa *Internship*, dan saya mencoba untuk *logIn* dengan *role* tersebut, ketika saya *logIn*, ternyata masih ada *error*, dikarenakan data yang ada di database saya belum terkoneksi dengan database sistem *WebApp JALA* jadi saya tidak bisa untuk mencoba *role* tersebut, setelah itu saya menanyakan ke mas farid tentang fitur Peran/*Roles* yang telah saya *update*, dan setelah selesai saya memeriksa semua pekerjaan yang telah saya kerjakan dan mengecek apakah masih ada yang salah atau tidak, lalu saya bertanya tentang kesalahan *role* yang saya buat, dikarenakan data yang berada pada tabel master hanya dapat diakses dan digunakan oleh tim software *JALA* saja, maka itu saya tidak diperkenankan untuk mengaksesnya, setelah selesai lalu saya diminta untuk melakukan *commit* ke *gitlab* dengan perubahan *code* yang telah saya kerjakan, karena pada fitur-fitur yang saya *update* sebelumnya ada beberapa yang belum di *commit*, jadi disini saya menjalankan syntax `git status` untuk mengecek fitur mana saja yang belum saya *commit*, lalu saya menjalankan syntax `git add .` untuk menyimpan semua perubahan yang telah saya lakukan ke dalam *staging*, setelah itu saya menjalankan syntax `git commit -m "Update fitur Roles, Akses dan Investors"`, yang dimana syntax `-m` digunakan untuk memberikan pesan bahwasanya fitur-fitur tersebut telah saya *update*.

Selasa 20 Agustus 2019 (09.20 – 17.10)

Pada Hari ketiga puluh dua saya diminta untuk mengecek semua tugas yang telah saya kerjakan dan saya diminta untuk melanjutkan melakukan *update* fitur user dengan cara dan langkah sama seperti hari sebelumnya sesuai dengan *Laravel-Generator*, file yang saya hapus yaitu File *Factories, Models, Filters, Policy, lang, view, Controller* dan *Feature*, setelah itu saya menjalankan syntax `php artisan make:model --model=Role RoleController`, setelah saya menjalankan syntax tersebut saya menemukan *error* yaitu model user yang telah saya hapus ternyata digunakan pada beberapa *class* jadi ketika saya menjalankan syntax tersebut muncul *error* dikarena file model user yang diperlukan tidak ada, jadi disini saya mengembalikan semua file user yang telah dihapus, dan mulai menghapus beberapa file yang berhubungan dengan controller tidak dengan model nya, yaitu file *Factories, Policy, lang, view, Controller* dan *Feature*, lalu saya menjalankan syntax `php artisan make:controller UserController`, setelah itu saya melakukan *update* pada file-file *controller* yang telah dihapus sesuai dengan langkah pada progres sebelumnya, setelah itu saya melanjutkan untuk melakukan pengetesan pada programnya dengan mengecek *code* dan alur program yang telah saya *update* pada fitur user, setelah selesai saya menanyakan ke mas farid tentang progres yang telah saya kerjakan selama ini, dan saya juga menanyakan tentang *error* yang saya alami, dan saya juga berbincang-bincang dengan mas Farid tentang hal yang sudah saya kerjakan, dan menanyakan lebih dalam lagi tentang *Laravel-generator* dan *framework Laravel* yang asli, disini juga saya diberikan penjelasan oleh mas Farid tentang penulisan *code*, manfaat *Laravel generator* yang dibuat, setelah itu saya diminta untuk melakukan *commit* ke *Gitlab* dengan langkah yang sama seperti progres sebelumnya.

Selasa 27 Agustus 2019 (09.30 – 17.35)

Pada hari ketiga puluh tiga saya belum diberikan tugas lanjutnya, dikarenakan saya lagi 3 hari saja melaksanakan Kerja Praktek di JALA, sembari mengisi waktu luang

saya belajar-belajar tentang *Github* dan *Gitlab* dengan beberapa teman-teman *internship* yang ada di JALA, disini saya belajar tentang penggunaan *Github* dan *Gitlab*, cara membuat *branch* dan juga belajar tentang penggunaan *GitFlow* pada *Github* dan *Gitlab*. Setelah itu saya belajar tentang *Laravel-Generator* dengan studi kasus yang berbeda, disini saya belajar tentang pembuatan aplikasi laundry dengan analisa dan soal yang saya cari di internet, setelah sedikit mencoba pembuatan aplikasi, lalu mas Farid menanyakan kepada saya tentang penggunaan *Vue.js*, lalu saya menjelaskan tentang penggunaan *Vue.js* yang saya pahami sesuai dengan yang saya pelajari di internet dan yang saya tanyakan ke mas Dama pada beberapa hari sebelumnya, disini saya diminta untuk membuat sebuah *CRUD* untuk *Vue.js* dengan 1 tabel yang diberikan oleh mas Farid, disini saya menggunakan *Vue.js* sebagai *Front-End* nya dan *Laravel* sebagai *Back-End* nya, langkah pertama yang saya lakukan yaitu menginstall *Laravel* dengan menggunakan `composer create-project -prefer-dist laravel/laravel Vue`, setelah menginstall *Laravel* saya mulai menjalankan `php artisan serve`, untuk memastikan bahwa *Laravel* sudah terinstall dengan benar, lalu saya melanjutkan untuk menginstall *Vue.js* yang dimana pada *Laravel* terbaru sudah langsung tersedia dependensi *Vue.js* pada *Laravel*, jadi saya cukup menjalankan `npm install vue`, setelah itu saya mulai menghubungkan database dengan *Laravel*, dengan menambahkan `DB_DATABASE = vue`, yang terletak pada file `.env` di *Laravel*, setelah itu saya mulai membuat satu *Back_End* atau *API CRUD* pada *controller*, dengan langkah pertama yang saya lakukan yaitu membuat sebuah *controller* dengan menggunakan `php artisan make:model -model=Crud CrudController`, setelah selesai saya melanjutkan untuk membuat satu fungsi *show* pada *Back_End Laravel*, setelah itu saya melakukan pengetesan dengan menggunakan *Postman*.

Rabu 28 Agustus 2019 (09.20 – 17.10)

Pada Hari ketiga puluh empat saya melanjutkan belajar tentang penggunaan *Vue.js* langkah yang saya lakukan yaitu membuat sebuah fungsi *create*, dan *delete* pada *Back-End Laravel*, dengan menggunakan syntax `DB::table('vue')->insert(['nama' => $request->name]);` yang dimana *code* tersebut digunakan untuk memasukan data nama kedalam database, disini saya menggunakan *array* karena data yang dimasukan lebih dari satu, setelah itu saya membuat sebuah fungsi *update* dan *delete* dengan langkah hampir sama dengan *insert* tetapi pada fungsi *delete* hanya menuliskan fungsi dengan syntax `DB::table('vue')->where('vue_id', $id->delete());` yang dimana syntax tersebut digunakan untuk menghapus data sesuai dengan *id* yang akan dihapus, setelah itu saya melakukan pengecekan semua fungsi yang telah saya dengan *Postman* sebelumnya disini saya sedikit menjelaskan apa itu *Postman*, ***Postman*** merupakan sebuah *tools* website *GUI* yang digunakan oleh para *developer* untuk membuat dan melakukan testing sebuah *API* atau *Back_End* yang dibuat, setelah itu saya lanjut pada pembuatan *view*, tetapi sebelumnya saya menanyakan ke mas Farid tentang progress yang telah saya kerjaka, dan mas Farid memberikan sedikit saran untuk penulisan fungsi dan nama variable agar menggunakan standar yang sudah ditentukan oleh Bahasa pemrograman PHP, setelah itu saya melanjutkan untuk membuat tampilannya pada *Vue.js*, langkah pertama yang saya lakukan yaitu seperti langkah sebelumnya dengan membuat sebuah file *component* pada *Vue.js* lalu melanjutkan membuat sebuah routing pada file *app.js*, setelah itu saya menanyakan ke mas Farid tentang pemanggilan fungsi *Back_end* ke tampilan *Vue.js*, disini mas Farid memberikan saya penjelasan tentang fungsi *axios* yang dimana fungsi ini yang akan digunakan untuk memanggil *API Back_End* dari *Laravel* ke *Vue*.

Senin 2 September 2019 (09.35 – 17.30)

Pada hari ketiga puluh lima yang dimana merupakan hari terakhir saya Kerja Praktek di JALA, seperti biasanya diadakan rapat oleh tim *software* JALA untuk membahas progress yang akan dikerjakan pada minggu ini, dikarenakan saya tidak mengikuti rapat pada minggu kemarin jadi disini saya menjelaskan sekaligus mempresentasikan progress yang saya kerjakan selama kerja praktek di JALA yang saya laksanakan kurang lebih 1 bulan, disini saya mempresentasikan tentang penggunaan *Laravel-Generator* dan progress yang telah saya kerjakan mulai dari pemahaman tentang *Laravel-Generator* progress *update* yang saya lakukan pada beberapa fitur di *WebApp* JALA, setelah itu saya memberikan *form* penilaian yang dinilai oleh mas Farid serta seluruh tim *software* JALA, setelah itu saya diajak sedikit berbincang bincang dengan teman-teman di JALA tentang kesan dan pesan saya selama kerja praktek di JALA, setelah itu diajak foto bersama dan saya berpamitan untuk menyelesaikan kerja praktek di JALA, sebelum pulang saya juga sempat untuk menanyakan ke mas Farid tentang apa yang telah saya kerjakan, dan mas Farid memberikan tanggapan bahwa saya sudah cukup bagus dibagian *coding* dan kerjasama dalam tim, tetapi disini mas Farid menekankan bahwa saya harus disiplin dalam bekerja serta memberikan nasehat untuk kedepannya harus bagaimana. Setelah itu saya berpamitan pulang kepada seluruh karyawan dan teman-teman *internship* yang masih di JALA.

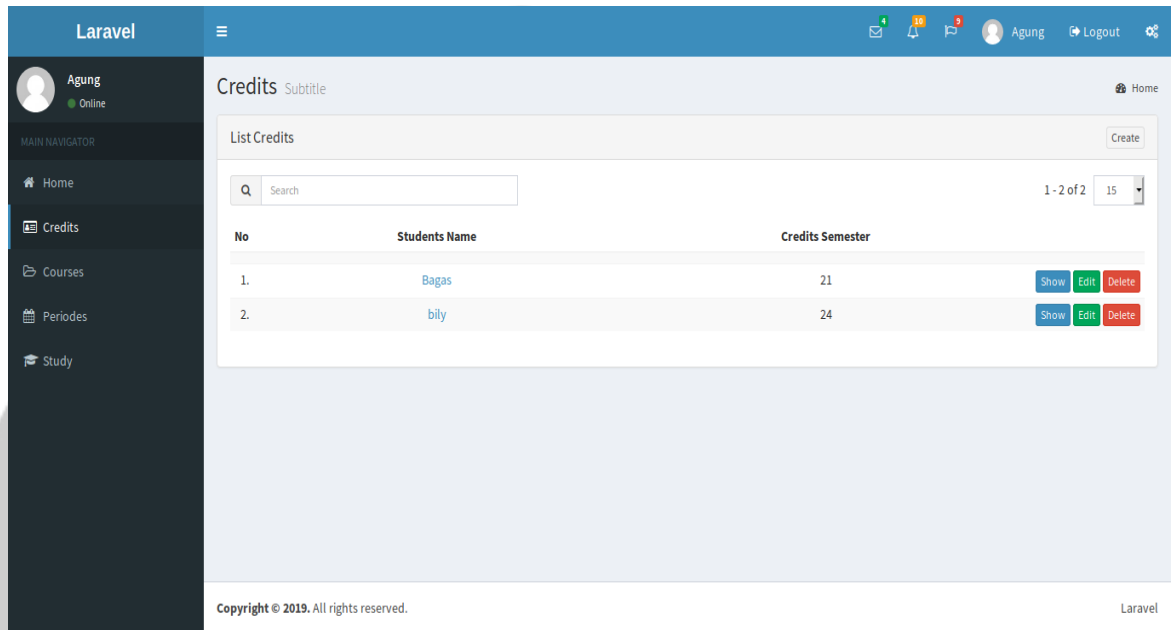
2.2 Hasil Pekerjaan Secara Umum

Pada kegiatan kerja praktek ini, saya membuat *update code* pada fitur-fitur yang berada di *WebApp JALA*. Halaman fitur yang saya *update* yaitu, dari Master Data ada Fitur Inventarisasi, Parameter, RFID, Species, TAG, Alat, Parameter Alat, Vendor, Produk, Produk Kategori, Operasional dan Aset, sedangkan dari data *Credentials* Fitur yang saya *update* yaitu Pengguna, Peran, Hak Akses, Fitur Akses, dan Invenstor, adapun dari semua fitur yang saya *update* dengan menggunakan *Laravel-generator-0.10* yang sudah didesain dan di *custume* oleh tim *software* dari JALA. Dan perbedaan dari sebelum di *update* dan sesudah di *update* itu terletak pada tampilan dari setiap fitur, yang dimana sebelum di *update* user hanya dapat mencari data berdasarkan semua *field* yang akan dicari, sedangkan setelah di *update* user bisa mencari data sesuai dengan *field* yang diinginkan dan juga ada beberapa perubahan *code*, untuk mempersingkat *code* dan mempermudah developer untuk manajemen *code* ketika ada kesalahan, dan fitur-fitur yang saya *update* hanya dapat diakses oleh tim *software* dan beberapa orang dari pihak JALA.

Selain itu, saya juga diberikan latihan tentang penggunaan *Laravel-Generator* yang sudah Dibuat oleh tim *software* JALA, yang dimana disini saya diminta untuk membuat sebuah sistem kampus sederhana. Sistem yang saya buat ini untuk latihan saya menggunakan dan mempelajari tentang *framework Laravel* generator yang sudah dibuat oleh tim *software* JALA, dan pada sistem saya membuat beberapa *CRUD* dengan *Laravel-Generator* dan membuat *role* user untuk membedakan akses setiap user ke beberapa tabel yang diberikan. Sistem yang dibangun dengan *framework Laravel* generator ini memiliki beberapa fitur, diantaranya adalah;

- Pengelolaan Data User
- Pengelolaan Data Courses
- Pengelolaan Data Study
- Pengelolaan Data Periode

2.3. Bukti Hasil Pekerjaan



Gambar 2.1 Tampilan Homepage Fitur Credits

Gambar 2.1 memperlihatkan tampilan *Homepage* fitur *credits*, yang dimana pada tampilan ini diakses oleh *role* yang bertipe admin, yang dimana *role* yang bertipe admin yang hanya bisa menambahkan dan mengedit data.

Create Credit

Student*

Bagas

Credit in This Semester*

24

Back Store Store & Create

Copyright © 2019. All rights reserved. Laravel

Gambar 2.2 Fitur Tambah Data Credits

Gambar 2.3 memperlihatkan tampilan tambah data *credits*, yang dimana user bertipe admin dapat menambahkan data dengan memilih nama mahasiswa dalam form *dropdown* dan mengisi jumlah SKS yang dapat diambil oleh mahasiswa pada semester ini.

Edit Credit

Credit in This Semester

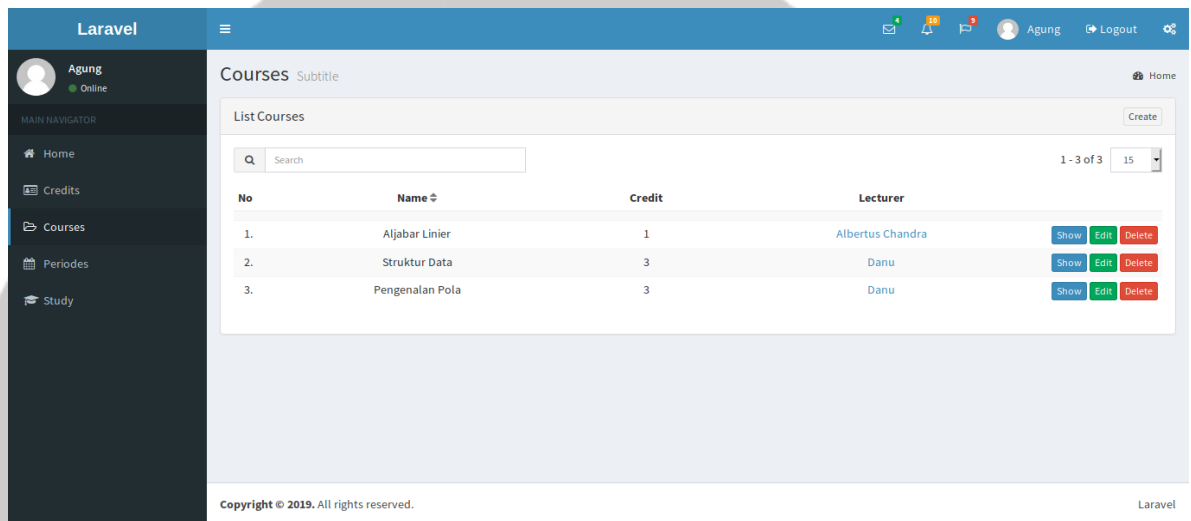
21

Back Delete Update

Copyright © 2019. All rights reserved. Laravel

Gambar 2.3 Fitur Edit Data Credits

Gambar 2.3 memperlihatkan tampilan edit data *credits*, yang dimana user bertipe admin dapat mengedit data dengan mengisi jumlah SKS yang dapat diambil oleh mahasiswa pada semester ini.



Gambar 2.4 Tampilan Homepage Fitur Courses

Gambar 2.4 memperlihatkan tampilan *Homepage* fitur *courses*, yang dimana pada tampilan ini diakses oleh *role* yang bertipe admin, yang dimana *role* yang bertipe admin yang hanya bisa menambahkan dan mengedit data. Dan pada fitur ini *role* yang bertipe dosen dapat juga mengedit dan menambahkan data.

Create Course

Courses Name*

Total Credit*

Lecturer Name*

[Back](#) [Store](#) [Store & Create](#)

Copyright © 2019. All rights reserved. Laravel

Gambar 2.5 Fitur Tambah Data Courses

Gambar 2.5 memperlihatkan tampilan tambah data *courses*, yang dimana user bertipe admin dapat menambahkan data dengan menginputkan nama matakuliah yang diampu oleh dosen, jumlah SKS matakuliah, dan memilih nama dosen pengampu matakuliah. Dan untuk user yang bertipe dosen hanya dapat menambahkan nama matakuliah dan jumlah SKS, karena data akan diinputkan sesuai dengan dosen yang *login*.

Edit Course

Courses Name*

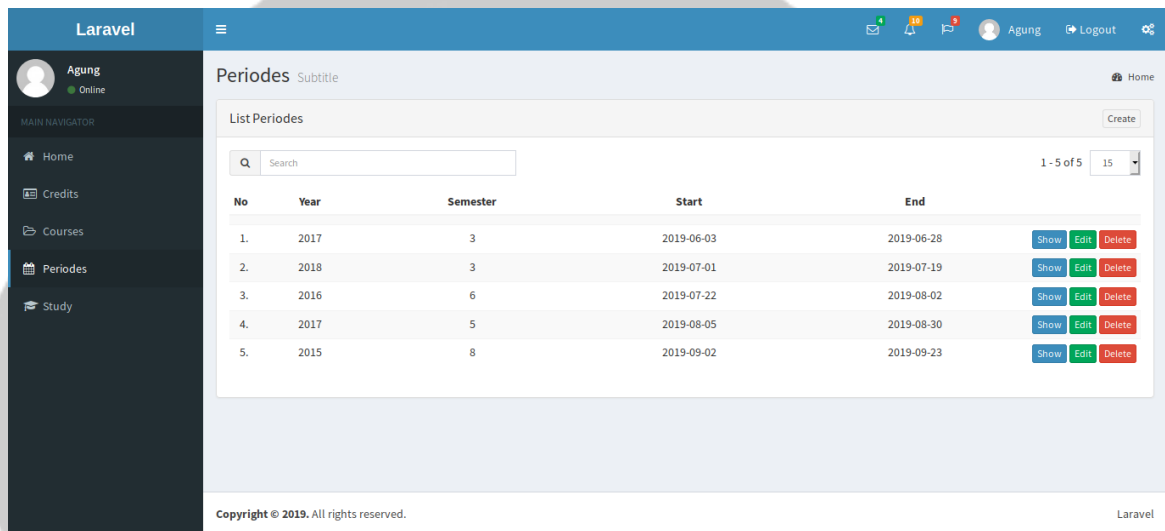
Total Credit*

[Back](#) [Delete](#) [Update](#)

Copyright © 2019. All rights reserved. Laravel

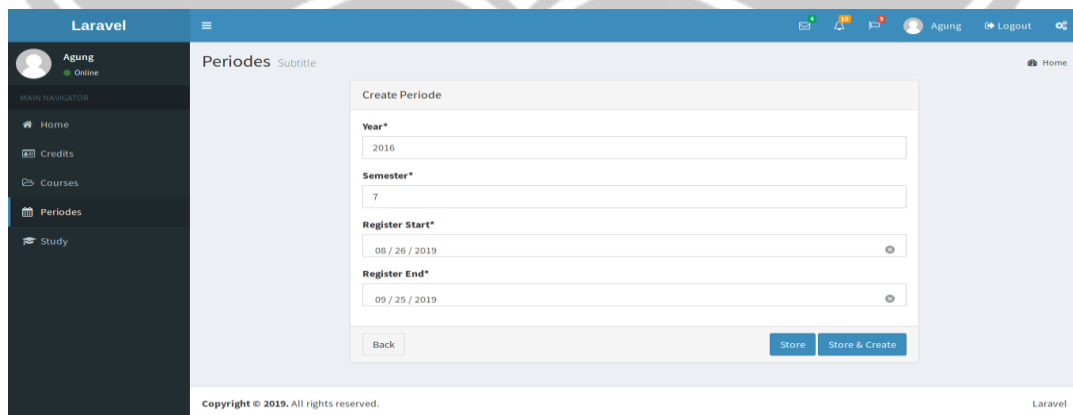
Gambar 2.6 Fitur Edit Data Courses

Gambar 2.6 memperlihatkan tampilan edit data *courses* yang dimana user bertipe admin dan dosen dapat mengedit data dengan menginputkan nama matakuliah dan jumlah SKS dari mata kuliah tersebut.



Gambar 2.7 Tampilan Homepage Fitur Periode

Gambar 2.7 memperlihatkan tampilan homepage fitur periode, yang dimana pada tampilan ini hanya diakses oleh user bertipe admin, karena hanya admin yang bisa mengedit dan menambahkan data pada fitur periode ini.



Gambar 2.8 Fitur Tambah Data Periode

Gambar 2.8 memperlihatkan tampilan tambah data periode, yang dimana pada tampilan ini hanya user yang bertipe admin yang dapat menambahkan data, dengan menginputkan tahun angkatan mahasiswa, semester, tanggal registrasi awal dan tanggal registrasi akhir.

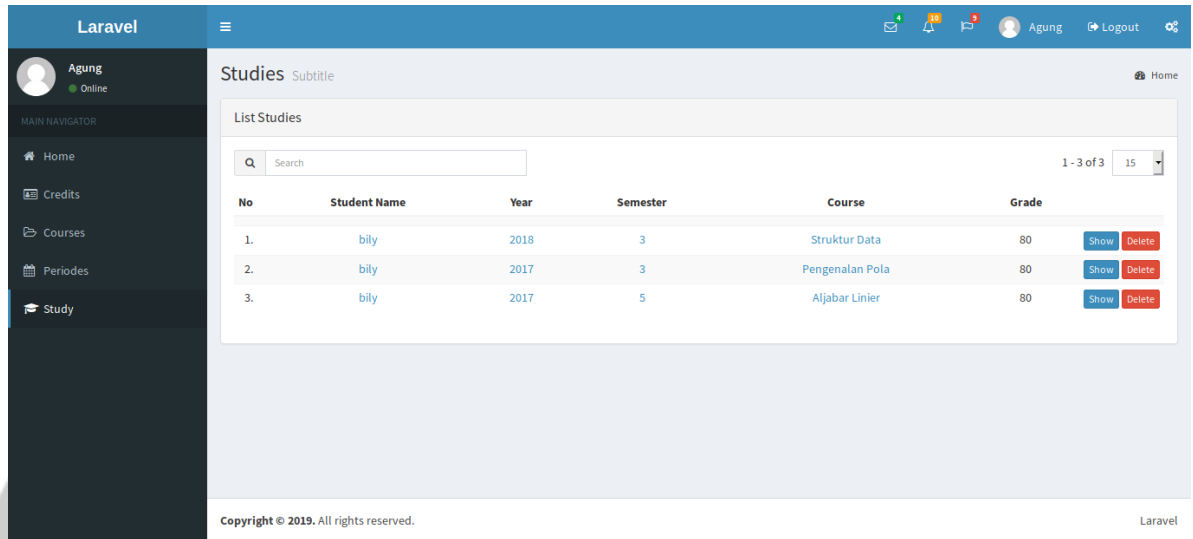
The screenshot displays a web application interface for editing a period. The main content area is titled 'Periodes Subtitle'. The 'Edit Periode' form includes the following fields and values:

Field	Value
Year*	2017
Semester*	3
Register Start*	06 / 03 / 2019
Register End*	06 / 28 / 2019

At the bottom of the form, there are three buttons: 'Back', 'Delete', and 'Update'.

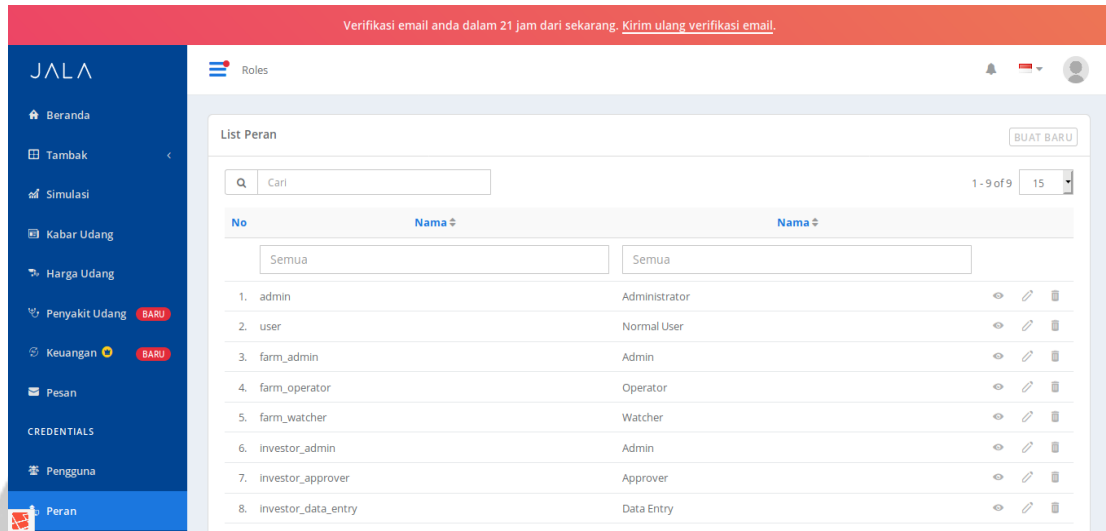
Gambar 2.9 Fitur Edit Data Periode

Gambar 2.9 memperlihatkan tampilan edit data periode yang dimana pada fitur ini yang dapat mengedit data hanya user yang bertipe admin, yang dimana data yang diedit yaitu tahun angkatan mahasiswa, semester, tanggal registrasi awal dan tanggal registrasi akhir.



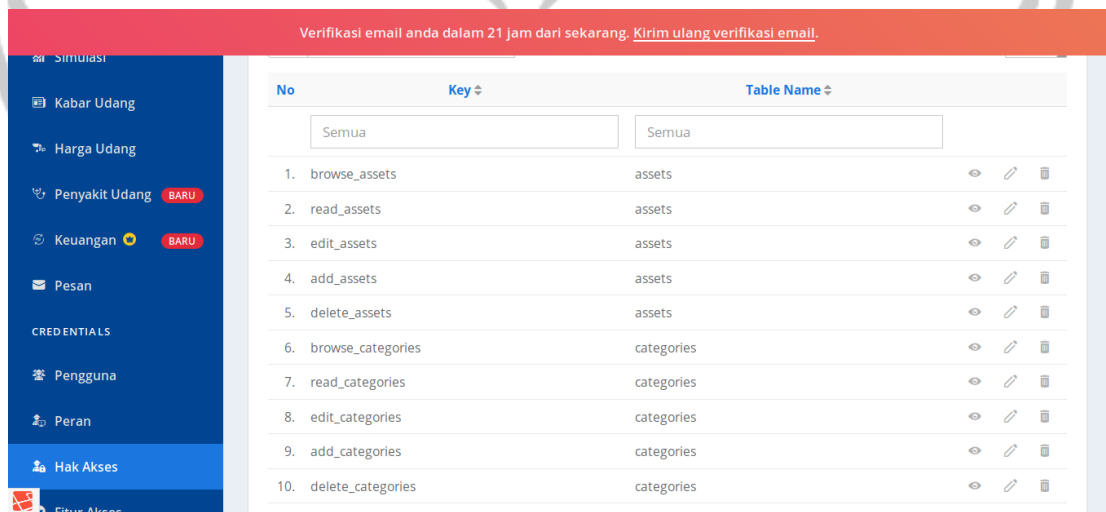
Gambar 2.10 Tampilan Homepage Fitur Study

Gambar 2.10 memperlihatkan tampilan Homepage pada fitur *study*, yang dimana disini user yang bertipe admin hanya dapat menghapus data dan user bertipe mahasiswa dan dosen dapat mengedit dan menambahkan data.



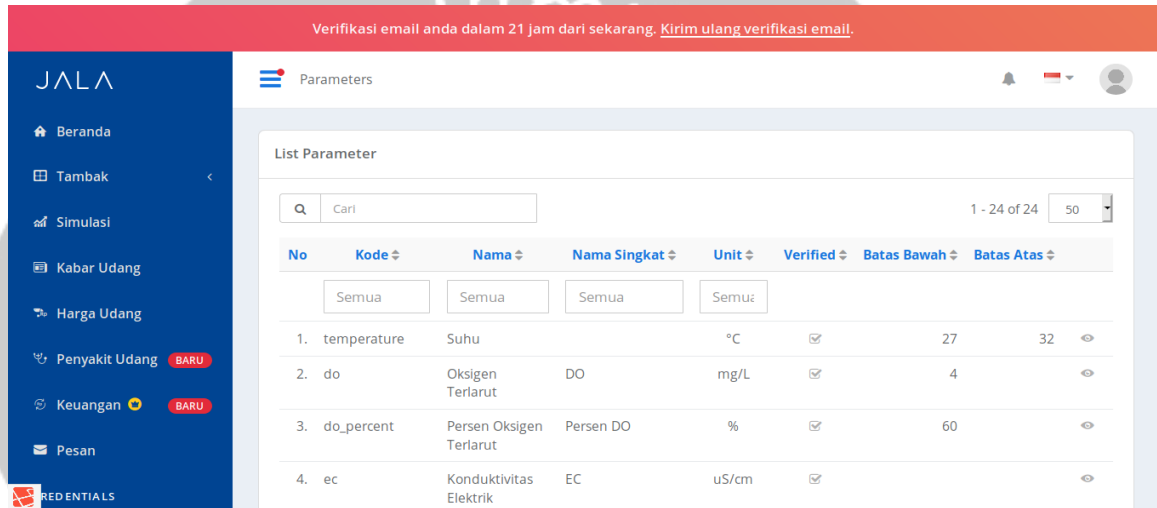
Gambar 2.11 Tampilan Homepage Fitur Peran

Gambar 2.11 memperlihatkan tampilan homepage fitur peran pada WebApp JALA yang sudah di *update* ke *Laravel-Generator-0.10*, yang dimana pada fitur ini user dapat mengakses dan menentukan nama peran yang dapat mengakses beberapa fitur yang berada di WebApp JALA.



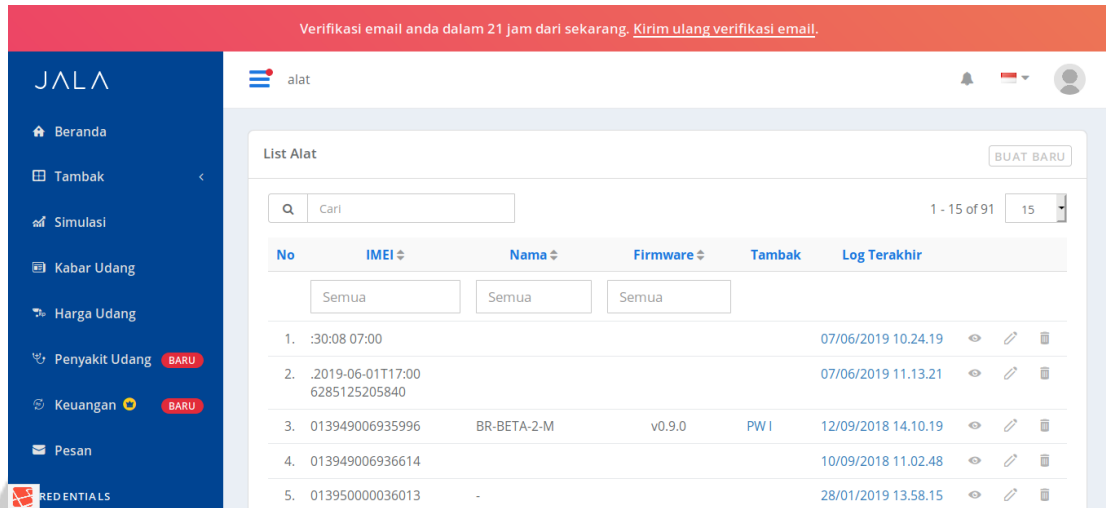
Gambar 2.12 Tampilan Homepage Fitur Hak Akses

Gambar 2.12 memperlihatkan tampilan *homepage* fitur hak Akses pada *WebApp* JALA yang dimana fitur ini telah diupdate ke *Laravel-Generator-0.10*, dan pada fitur ini admin dapat memberikan hak akses kepada user yang bisa mengakses beberapa fitur yang ada di *WebApp* JALA.



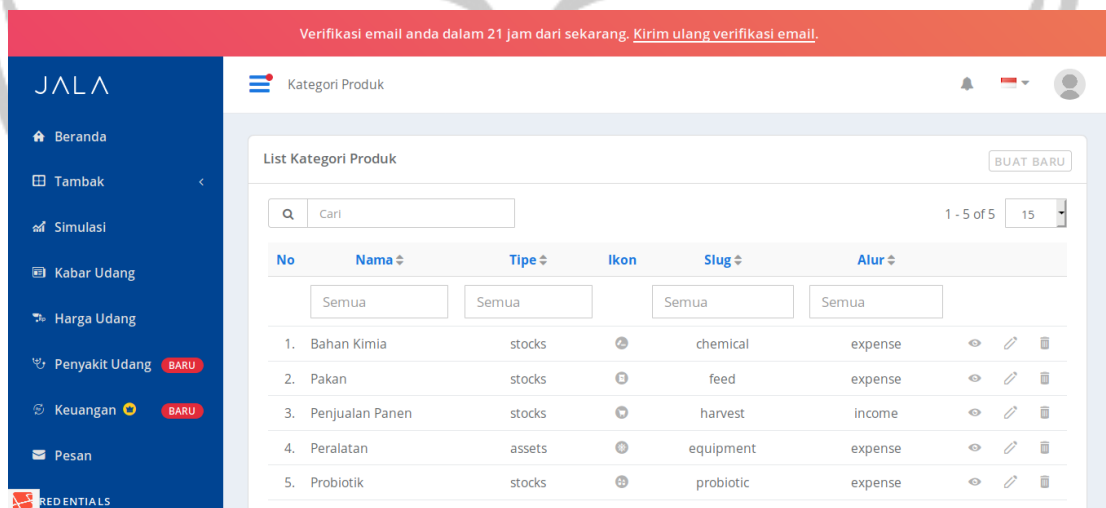
Gambar 2.13 Tampilan Homepage fitur Parameter

Gambar 2.13 memperlihatkan tampilan fitur parameter yang berada di *WebApp* JALA, yang dimana fitur ini telah di update ke *Laravel-Generator-0.10*, pada fitur ini admin dapat mengelola data yang dikirim oleh alat JALA yang dikirim langsung ke *WebApp* JALA, yang dimana data parameter disini akan dikelola dan di cek oleh admin ketika ada *error* pada alat JALA.



Gambar 2.14 Tampilan Homepage Fitur Alat

Gambar 2.14 memperlihatkan tampilan fitur alat pada *WebApp JALA*, yang dimana fitur ini telah diupdate ke *Laravel-Generator-0.10*, pada fitur ini admin dapat mengelola data alat-alat JALA, yang berisi tentang *IMEI*, *Nama*, *Firmware* yang digunakan, dan diletakan pada tambak yang mana.



Gambar 2.15 Tampilan Homepage Fitur Kategori Produk

Gambar 2.15 memperlihatkan tampilan *homepage* fitur kategori produk, yang dimana fitur ini telah di *update* ke *Laravel-Generator-0.10*, pada fitur ini admin dapat mengelola data kategori produk yang dimana berisi list nama produk, produk masuk kategori mana, ada aset, stock dan beberapa kategori lainnya.



BAB III

HASIL PEMBELAJARAN

3.1 Manfaat Kerja Praktek

Kerja Praktek memberikan banyak manfaat khususnya kepada diri saya sendiri, karena pada saat Kerja Praktek saya bisa menerapkan ilmu-ilmu yang telah didapatkan pada masa perkuliahan dan manfaat yang lainya menurut saya yaitu terkait dengan pemahaman saya terhadap dunia kerja, dengan melaksanakan Kerja Praktek, saya dapat memahami dan mengetahui dunia kerja IT sesungguhnya dan teknologi-teknologi apa saja yang banyak digunakan oleh perusahaan-perusahaan yang memanfaatkan IT pada proses bisnisnya. Dengan mengetahui berbagai hal ini, saya pribadi merasa lebih siap dan nyaman ketika nanti langsung terjun ke dunia kerja.

Saya juga belajar bagaimana saya harus beradaptasi dengan lingkungan yang baru, bekerja sama dalam tim serta berdiskusi dan saling bertukar pikiran terkait berbagai masalah yang dihadapi. Di PT. Atnic Ekotekno Wicaksana sendiri terdapat 2 rapat rutin setiap minggunya untuk tim *software*. Rapat dilaksanakan pada hari senin dan jumat. Rapat ini memberikan saya pemahaman yang lebih tentang pentingnya komunikasi antara anggota tim satu dengan yang lainnya.

Melalui kerja praktek ini, saya banyak mempelajari tentang *tools-tools* baru yang tidak saya dapatkan pada saat perkuliahan, contohnya menggunakan *Laravel* dan *Vue.js* untuk membuat sebuah sistem sederhana yang dimana saya banyak mempelajari tentang *tools-tools* baru dan *library-library* yang ada di *Vue.js* walaupun pada saat perkuliahan saya mendapatkan beberapa materi tentang pembuatan web, tetapi dengan kerja praktek ini saya menjadi lebih memahami tentang materi materi tersebut dan saya juga mendapatkan banyak hal-hal yang baru, yang dimana bisa saya manfaatkan untuk kesempatan yang akan datang.

3.2 Penerapan Ilmu Dalam Kerja Praktek

Kerja Praktek ini dapat terlaksana dengan baik tidak terlepas dari ilmu pengetahuan yang telah saya dapatkan pada proses perkuliahan di Program Studi Teknik Informatika, Universitas Atma Jaya Yogyakarta. Hal yang pertama bagaimana saya harus bekerja dalam tim, sejak awal perkuliahan dengan adanya berbagai tugas kelompok maupun berbagai partisipasi dalam organisasi dan event tertentu, saya mendapatkan banyak pengalaman yang berharga. Selain itu, saya juga menerapkan berbagai macam ilmu yang sudah saya pelajari pada beberapa mata kuliah untuk mengerjakan *project* pada saat kerja praktek ini.

Mata Kuliah yang memiliki peran terbesar yaitu P3L (*Project Pengembangan Perangkat Lunak*). P3L merupakan mata kuliah wajib yang saya ambil pada semester 6, yang dimana pada matakuliah ini semua mahasiswa yang mengambil diwajibkan untuk membuat sistem informasi terkait studi kasus yang telah ditentukan. Pada matakuliah ini saya mempelajari banyak sekali tentang software development yang sebelumnya belum pernah saya pelajari. Mulai dari *project management*, *GIT*, hingga pembuatan programnya, yang dimana saya mendapatkan tugas untuk membuat sebuah website dengan menggunakan *framework Laravel*, pada matakuliah ini saya mempelajari tentang penggunaan *framework Laravel* dimana *framework* tersebut saya gunakan kembali pada saat pelaksanaan kerja praktek.

Mata Kuliah lain yang ilmunya diterapkan selama saya melaksanakan kerja praktek adalah Pemrograman Web. Mata Kuliah ini saya ambil pada semester 3, pada matakuliah pemrograman web, saya mempelajari hal-hal mendasar terkait programming web, pada saat ini saya juga mempelajari *framework php* yang terbaru pada saat ini yaitu *framework Yii*, dan juga pada mata kuliah ini saya belajar tentang dasar-dasar pemrograman web seperti, *HTML*, *CSS*, *JQuery*, *PHP*, materi-materi yang saya dapatkan pada saat perkuliahan ini tentang menjadi modal baik bagi saya

untuk mengerjakan *project* yang diberikan pada saat kerja praktek, meskipun saya harus mempelajari materi-materi tersebut secara lebih dalam.



BAB IV

KESIMPULAN

Kerja praktek yang saya laksanakan dari tanggal 2 Juli 2019 sampai 2 September 2019 di PT. Atnic Ekotekno Wicaksana telah berjalan dengan lancar. Dalam pelaksanaan kerja praktek saya bergabung di tim *software*. Pada pelaksanaan kerja praktek ini saya mendapatkan 2 *project* untuk membuat sebuah sistem kampus sederhana dengan menggunakan *Laravel* generator dan membuat *update* pada Fitur data *Credentials* dan Data Master pada *WebApp* JALA.

Pelaksanaan kerja praktek tentunya memberikan banyak manfaat khususnya bagi diri saya sendiri. Hal ini dikarenakan saya sebagai mahasiswa Teknik Informatika yang akan melanjutkan bekerja pada perusahaan-perusahaan yang memiliki department IT, sehingga sangat penting bagi saya untuk mendapatkan pengetahuan dan juga pengalaman mengenai bagaimana dunia kerja di bidang IT yang sesungguhnya. Dan juga saya menjadi lebih mengetahui tentang teknis mengenai berbagai macam teknologi yang digunakan di perusahaan, hingga betapa pentingnya koordinasi, kerjasama, dan juga komunikasi antara anggota tim, sehingga sangat penting bagi saya untuk mempelajari berbagai macam teknologi terbaru.